

**UNIVERSIDAD AUTÓNOMA DE MADRID**

**ESCUELA POLITÉCNICA SUPERIOR**



**TRABAJO FIN DE MÁSTER**

**TECNOLOGÍAS PARA LA ADAPTACIÓN AL USUARIO  
EN INTERFACES CEREBRO-MÁQUINA BASADOS EN  
POTENCIALES VISUALES EVOCADOS**

**Máster Universitario en Investigación e Innovación en  
Tecnologías de la Información y las Comunicaciones**

**Autor: Pérez Martín, Aarón  
Tutor: Varona Martínez, Pablo**

**Grupo de Neurocomputación Biológica (GNB)  
Dpto. Ingeniería Informática**

**FECHA: Septiembre, 2019**

# **TECNOLOGÍAS PARA LA ADAPTACIÓN AL USUARIO EN INTERFACES CEREBRO-MÁQUINA BASADOS EN POTENCIALES VISUALES EVOCADOS**

**Autor: Pérez Martín, Aarón  
Tutor: Varona Martínez, Pablo**

**Grupo de Neurocomputación Biológica (GNB)  
Dpto. Ingeniería Informática  
Escuela Politécnica Superior  
Universidad Autónoma de Madrid  
Septiembre, 2019**

## Agradecimientos

Agradecimientos a mi familia y amigos por el apoyo dedicado a pesar de la distancia, a los compañeros del GNB por su participación en este proyecto y a mi tutor Pablo por su dedicación, sabiduría y paciencia durante todo el proceso. Gracias también por dejarme participar en este apasionante proyecto con las interfaces cerebro máquina. He aprendido mucho más de lo que habría imaginado y seguiré avanzando en este nuevo camino.

Finalmente y el más importante, quería dedicar un especial agradecimiento a mi pareja Sarah Franzen por el amor y cariño demostrado tanto en los buenos momentos como en los difíciles. A pesar de la distancia, fue para mí una motivación incombustible y que sin ella no habría podido continuar.

Aarón Pérez Martín

Septiembre, 2019

## **Resumen**

En los últimos años la aplicación de nuevas tecnologías en el campo de la Neurociencia y la Medicina en general ha tenido y sigue teniendo un importante papel en su desarrollo. Gracias a ello la comunicación entre los sistemas electrónicos y las personas se ha visto incrementado. Por ello, las interfaces cerebro máquina (BCI), que son sistemas electrónicos capaces de registrar ondas cerebrales mediante un electroencefalograma (EEG), pueden ser de gran utilidad a personas con deficiencias motoras. Muchos son los esfuerzos de grupos científicos por mejorar estos sistemas no invasivos y que son considerados como la pieza clave para el desarrollo de prótesis. A pesar de los esfuerzos, no se consigue alcanzar la robustez necesaria como para que su implementación sea de uso diario.

Este proyecto se va a centrar en un tipo de señales EEG denominadas potenciales evocados visualmente de estado estacionario (SSVEP) y que son una respuesta del cerebro a un estímulo visual externo que parpadea a una determinada frecuencia y fase. Gracias al trabajo previo en el Grupo de Neurocomputación Biológica (GNB), se va utilizar un sistema BCI basado en estímulos mediante LEDs.

La aportación de este trabajo será la creación de una plataforma Software que permita a un usuario realizar estimulaciones y la detección de señales SSVEPs para diversos estudios científicos que consideran la adaptación al usuario. Para ello, el usuario podrá determinar cuáles son las mejores frecuencias, fases y electrodos del sujeto a estudiar, es decir, se desea descubrir la mejor adaptación al usuario usando un sistema BCI seleccionando estos parámetros. Para facilitar su integración se ha optado por tecnologías recientes, QT y Python, estimulación con LEDs y una metodología de detección de SSVEPs conocida como el Análisis de la correlación Canónica (CCA). Finalmente un sistema de puntuación implementado en esta plataforma permitirá al usuario de determinar si las adaptaciones son compatibles ente sí en el marco de estudio.

## **Palabras clave**

Interfaces cerebro máquina (BCI), electroencefalografía (EEG), SSVEP, Análisis de la correlación Canónica (CCA), Python, Adaptación personalizada en BCIs.



## **Abstract**

In recent years the application of new technologies in the field of Neuroscience and Medicine has had and still has an important role in their scientific development. Thus, communication between electronic systems and humans has increased. Brain machine interfaces (BCI), which are electronic systems capable of recording brain waves using an electroencephalogram (EEG), can be very useful for people with motor impairments. Many are the efforts of scientific groups to improve these non-invasive systems and which are considered as a key step for the development of prosthetic devices. Despite these efforts, it is not possible yet to achieve the necessary robustness for their implementation in a daily use.

This project will focus on a type of EEG signals known as visually evoked steady state potentials (SSVEP), which are a response of the brain to an external visual stimulus flickering at a given frequency and phase. Thanks to previous work at the Biological Neurocomputing Group (GNB), a BCI system based on visual stimulus using LEDs will be used.

The contribution of this work will be the creation of a Software platform that allows a user to perform stimulation and the detection of SSVEPS signals for various scientific studies. With this platform, the user can determine the best frequencies, phases and electrodes for each subject to be studied. In other words, the system aims for the best adaptation to the user using a BCI system by selecting these parameters. To facilitate its integration, we have opted for recent technologies, QT and Python, LED stimulation, and a methodology known as the Canonical Correlation Analysis (CCA) for SSVEP detection. Finally, a scoring system allows the user to determine if the adaptations are compatible with each other within this study framework.

## **Keywords**

Brain machine interfaces (BCI), electroencephalography (EEG), SSVEP, Canonical Correlation Analysis (CCA), Python, BCI personalized adaptation.



# Índice de Contenido

<b>1</b>	<b>Introducción</b>	<b>3</b>
1.1	Motivación . . . . .	3
1.2	Objetivos . . . . .	4
1.3	Estructura de la Memoria . . . . .	5
<b>2</b>	<b>Estado del Arte</b>	<b>7</b>
2.1	Actividad cerebral humana . . . . .	7
2.1.1	Tipos de registros de actividad del sistema nervioso . . . . .	7
2.1.2	Monitorización e interpretación de señales del sistema nervioso . . . . .	9
2.2	Interfaces cerebro máquina (BCI) . . . . .	11
2.2.1	Definición de BCI . . . . .	11
2.2.2	Tipos de BCI . . . . .	12
2.2.2.1	Dependientes vs. Independientes . . . . .	13
2.2.2.2	Invasivo vs. No Invasivo vs semi Invasivo . . . . .	13
2.2.2.3	Síncrono vs. Asíncrono . . . . .	13
2.3	Interfaces cerebro máquina (BCI) basados en EEG con señales SSVEPs . . . . .	13
2.3.1	Motivo de elección de los SSVEPs . . . . .	13
2.3.2	Etapas del sistema . . . . .	14
2.3.2.1	Estimulación . . . . .	14
2.3.2.2	Adquisición . . . . .	14
2.3.2.3	Procesado . . . . .	15
2.3.2.4	Aplicación de comandos . . . . .	16
2.4	Estado del arte en detección y análisis de SSVEPs . . . . .	17
2.4.1	Tabla resumen del estado del arte de BCIs que utilizan SSVEPs . . . . .	19
2.5	Motivación del trabajo . . . . .	20
<b>3</b>	<b>Diseño del sistema</b>	<b>23</b>
3.1	Objetivos en el diseño . . . . .	23
3.1.1	Simplicidad vs. Complejidad . . . . .	23
3.1.2	Usabilidad vs. Funcionalidad . . . . .	23
3.1.3	Fácil de aprender vs. Fácil de usar . . . . .	23
3.1.4	Modularidad y Reusabilidad . . . . .	24
3.2	Diseño de la plataforma . . . . .	24
3.2.1	Lenguajes de programación . . . . .	24
3.2.2	Interfaz gráfica de usuario . . . . .	25
3.3	Diseño de etapas . . . . .	26
3.3.1	Estimulación . . . . .	26
3.3.1.1	Tipos de estimulación lumínica para evocar SSVEPs . . . . .	27
3.3.1.2	Tipos de señales de estimulación . . . . .	28
3.3.2	Adquisición de EEG . . . . .	28
3.3.3	Procesado . . . . .	29
3.3.3.1	Elección de metodología de detección de SSVEPs . . . . .	29
3.3.3.2	Detección de mejores frecuencias . . . . .	31

3.3.3.3	Selección de mejores fases . . . . .	31
3.3.3.4	Selección de mejores electrodos . . . . .	31
3.3.4	Aplicación de comandos y evaluación del rendimiento . . . . .	32
<b>4</b>	<b>Desarrollo del sistema</b>	<b>33</b>
4.1	Interfaz gráfica de usuario (GUI) . . . . .	33
4.1.1	Metodología en el desarrollo . . . . .	33
4.1.2	Interfaz gráfica . . . . .	33
4.1.3	Secciones en la interfaz gráfica de usuario . . . . .	34
4.2	Configuración del BCI . . . . .	35
4.2.1	Estructura de la plataforma . . . . .	35
4.2.2	Estructura de ficheros . . . . .	36
4.2.3	Sistema de estimulación . . . . .	36
4.2.3.1	LEDs . . . . .	36
4.2.3.2	Arduino UNO . . . . .	37
4.2.3.3	Sistema de estimulación integrado . . . . .	37
4.2.4	Sistema de grabación . . . . .	37
4.2.4.1	Electrodos secos activos . . . . .	38
4.2.4.2	Malla de electrodos . . . . .	38
4.2.4.3	Dispositivos electrónicos . . . . .	39
4.3	Problemas solventados . . . . .	40
4.3.1	Globales . . . . .	41
4.3.2	En implementación de algoritmos: . . . . .	41
4.3.3	En Interfaz gráfica . . . . .	42
4.3.4	En validación . . . . .	42
<b>5</b>	<b>Resultados</b>	<b>43</b>
5.1	Resultados de validación de la plataforma . . . . .	43
5.1.1	Control de la estimulación LED . . . . .	43
5.1.1.1	Secuencia de comandos de estimulación . . . . .	43
5.1.2	Validación de la estimulación y el registro simultáneo de EEG . . . . .	44
5.1.2.1	Señal de estímulo . . . . .	45
5.1.2.2	Validación en el registro del estímulo en la señal de EEG . . . . .	45
5.1.2.3	Análisis de la respuesta al estímulo en la señal de EEG . . . . .	45
5.1.2.4	Análisis del registro EEG en frecuencias . . . . .	46
5.1.3	Validación de detección señales SSVEPs con un estímulo . . . . .	47
5.1.4	Validación de detección señales SSVEPs con cuatro estímulos . . . . .	47
5.1.5	Validación de las distintas etapas de un experimento . . . . .	49
5.1.5.1	Datos de sesión y usuario . . . . .	49
5.1.5.2	Selección de Modelo de localización de electrodos . . . . .	49
5.1.5.3	Validación en la selección de frecuencias de estímulo y trials . . . . .	50
5.1.5.4	Validación en la selección de Frecuencia, rango de fases y trials . . . . .	51
5.1.5.5	Validación con los mejores parámetros . . . . .	52
5.1.6	Validación en la extracción de datos del experimento . . . . .	54
5.1.7	Modalidad en las pruebas . . . . .	54

<b>6 Conclusiones y Futuro Trabajo</b>	<b>57</b>
6.1 Conclusiones . . . . .	57
6.2 Futuro Trabajo . . . . .	58
6.2.1 Interfaz gráfica de usuario . . . . .	58
6.2.2 Funcionalidad . . . . .	58
<b>Bibliografía</b>	<b>63</b>
<b>A Glosario</b>	<b>65</b>
<b>B Apéndice</b>	<b>67</b>
B.1 Manual de instalación del entorno . . . . .	67
B.1.1 API g.USBAmP . . . . .	67
B.1.2 Python 3.6.8 . . . . .	68
B.1.3 Librerías necesarias para la plataforma . . . . .	68



# Lista de Figuras

2.1	Clasificación de señales cerebrales para su uso en BCIs.s . . . . .	8
2.2	Imagen de un sistema BCI . . . . .	12
2.3	Imagen del modelo 10-20 de Electrodo EEG . . . . .	15
2.4	Imagen comparativa tipos de Neurofeedback en BCIs . . . . .	16
3.1	Diseño interfaz gráfica de usuario . . . . .	26
3.2	Configuración del estímulo . . . . .	26
3.3	Lanzamiento del estímulo . . . . .	27
3.4	Imagen del Análisis de la Correlación Canónica (CCA) . . . . .	29
4.1	Interfaz gráfica de usuario (GUI) . . . . .	34
4.2	Sistema de estimulación LED . . . . .	37
4.3	Imagen de Arduino Uno . . . . .	37
4.4	Dispositivo g.USBamp . . . . .	38
4.5	Imágenes de los tipos de electrodos usados . . . . .	38
4.6	Localización de electrodos . . . . .	39
4.7	Imágenes de la malla de electrodos . . . . .	39
4.8	Preamplificador de electrodos . . . . .	40
4.9	Amplificador de señal . . . . .	40
5.1	Secuencias de comandos de estimulación . . . . .	44
5.2	Función desarrollada en Python que ejecuta comandos de Linux . . . . .	44
5.3	Señales de estimulación del Arduino . . . . .	45
5.4	Ejemplo de validación del registro de EEG en el electrodo OZ . . . . .	45
5.5	Diferencia entre señales procedente de electrodos cercanos. . . . .	46
5.6	Sincronización señal estímulo del Arduino y la registrada en el EEG. . . . .	46
5.7	Demostración de detección visual de la frecuencia de estímulo . . . . .	47
5.8	Ejemplo de la problemática que se plantea en algunos casos para decodificar el estímulo en la señal de EEG cuando hay varios picos de amplitud parecida en el análisis de la transformada de Fourier . . . . .	47
5.9	Imagen de detección automática de señales SSVEPs desde la plataforma Software, un estímulo a la vez. . . . .	48
5.10	Imagen de detección automática de señales SSVEPs desde la plataforma Software, cuatro estímulos a la vez. . . . .	48
5.11	Interfaz: sección de usuario y sesión . . . . .	49
5.12	Interfaz: sección de modelo . . . . .	50
5.13	Interfaz: sección de frecuencia y electrodos . . . . .	50
5.14	Interfaz: resultado de la frecuencias y electrodos . . . . .	50
5.15	Interfaz: sección de fase . . . . .	52
5.16	Interfaz: resultado de la fase . . . . .	52
5.17	Interfaz: sección de validación . . . . .	52
5.18	Interfaz: resultado de la validación . . . . .	53
5.19	Interfaz: sección de validación alternativa . . . . .	53

5.20	Interfaz: resultado de la validación alternativa . . . . .	54
5.21	Interfaz: exportación en JSON de los parámetros del experimento . . .	55



# Lista de Tablas

2.1	Tabla resumen de tipo de señales cerebrales y sus características (Ramadan and Vasilakos, 2017). . . . .	9
2.2	Tabla resumen de tipos de ritmos cerebrales (Ramadan et al., 2015). . .	10
2.3	Tabla resumen de técnicas de registro de actividad cerebral (Ramadan and Vasilakos, 2017). . . . .	11
2.4	Tabla resumen del Estado del Arte de técnicas usadas en detección de SSVEPs . . . . .	20
2.5	Tabla resumen del Estado del Arte y los resultados . . . . .	21
5.1	Tabla resultado de las mejores frecuencias . . . . .	51
5.2	Tabla resultado de los mejores electrodos . . . . .	51
5.3	Tabla resultado de las mejores frecuencias estimuladas en conjunto . .	53
5.4	Tabla resultado de frecuencias en una configuración alternativa. . . . .	54



## Capítulo 1

# Introducción

### 1.1 Motivación

Desde que apareció el concepto de interacción entre persona y ordenador, se ha estado gestando un sinfín de artefactos que han ayudado al usuario a realizar las tareas deseadas (Myers, 1998). Esto supuso y sigue siendo un avance en el progreso humano al poder controlar tanto los propios ordenadores como otras máquinas a través de éstos.

Poco a poco se fueron distinguiendo entre componentes Hardware, que podría ser cualquier elemento físico conectado al ordenador de alguna forma, y componentes Software, que se entiende como cualquier pieza de código ejecutada en el ordenador, y que ambas contribuyen a la realización de la meta requerida por el usuario.

Hoy en día existe lo que se conoce como interfaces cerebro-máquina (BCI, por sus siglas en Inglés) y que se desarrollan en un ámbito multidisciplinar que combina principalmente las ramas de Electrónica, Ingeniería del Software y Neurociencia (Nicolas-Alonso and Gomez-Gil, 2012). El concepto es relativamente reciente (Wolpaw et al., 2002), y cuya razón de existir es la de registrar actividad cerebral para posteriormente controlar algún tipo de dispositivo externo a través de los ordenadores. Esto es posible porque el cerebro es un órgano capaz de producir ondas cerebrales que son registradas por electrodos y posteriormente procesadas por un ordenador para acometer alguna función, como podría ser el control de una silla de ruedas o un robot. En los últimos años los BCIs están teniendo un fuerte auge. Esto es debido a la aparición de nuevas técnicas, tanto de registro como de decodificación y clasificación de actividad neuronal (Waytowich et al., 2018; Attia et al., 2018), y que se combinan con los BCIs para intentar lograr una mejora en la precisión.

Las interfaces cerebro-máquina se utilizan fundamentalmente para proporcionar un medio de comunicación a personas con altas discapacidades. A pesar de los esfuerzos por lograr una alta efectividad en los BCIs, la realidad hoy en día es que no existe ninguna metodología que garantice la robustez necesaria como para que estos sistemas puedan ser implantados en las vidas cotidianas. Más allá de las múltiples aplicaciones para las que pueden ser usados estos sistemas, existen un gran número de potenciales usuarios que urgen una solución a las deficiencias en su movilidad o desordenes en el sistema nervioso motor, el cual es el encargado de coordinar movimientos.

Las causas de la poca efectividad lograda hasta la fecha son muchas. Una de ellas es la dificultad para decodificar la voluntad de realizar un movimiento en el contexto de toda la actividad que se registra en el cerebro. La implementación de las interfaces cerebro máquina es también un problema muy común ya que la electrónica es sensible a interferencias internas, por los propios músculos y movimientos internos,

y externas, como son la luz y las vibraciones, además de los típicos problemas de ajustes de los electrodos a la superficie de la cabeza.

Otra causa es la variabilidad de respuesta entre usuarios con el mismo sistema y mismos parámetros. Este problema también se repite con un mismo usuario, variando así su respuesta en el tiempo. Uno de los retos más grandes en estos sistemas es el de solucionar la incapacidad que tiene cierto porcentaje de la población, entre un 10% y 30%, a usar estos sistema y que se conoce como BCI Illiteracy (Vidaurre and Blankertz, 2010; Choi et al., 2017).

Este proyecto pretende avanzar en la mejora de las interfaces cerebro máquina (BCI). Para ello se analizarán las señales del electroencefalograma (EEG) que emite el cerebro humano y en concreto se registrarán los potenciales visuales evocados en estado estacionario (SSVEP, siglas en inglés) como respuesta a estímulos parpadeantes externos como medio para evocar actividad cerebral fácilmente detectable en un BCI. La principal motivación en este trabajo es la de proporcionar una mayor adaptación al usuario de los interfaces cerebro-máquina que utilizan potenciales visuales evocados. Esta adaptación combinará la elección personalizada de parámetros de estímulo como la frecuencia y la fase, así como la selección de electrodos en el sistema de registro y de la metodología de detección de los potenciales evocados.

Los principales beneficiarios de este estudio son aquellas personas que por el motivo que fuese tienen serias dificultades motoras o una degeneración cognitiva acusada y que necesiten comunicarse mediante algún mecanismo alternativo a los que requieren movimiento muscular.

## 1.2 Objetivos

Para llevar a cabo este trabajo se han seguido una serie de hitos que proporcionan una forma de trabajo escalonada que permite ir avanzando en su desarrollo.

Como ya se adelantaba en el apartado anterior, el trabajo centra su atención en la detección de señales SSVEPs usando interfaces cerebro máquina (BCI). En concreto, se presenta el desarrollo de una plataforma Software que permita la estimulación y la detección de señales EEG de una forma fácil e intuitiva adaptada a cada usuario.

Los objetivos del trabajo se resumen en los siguientes puntos:

- Realizar un estudio de los algoritmos más actuales en la detección de señales SSVEPs y los tipos de estimulación más comunes, para posteriormente determinar cuál se va a integrar en este trabajo.
- Desarrollar una plataforma Software Open Source que permita la estimulación personalizada mediante LEDs y la detección de señales SSVEPs generadas por el usuario del BCI mediante el método del Análisis de la Correlación Canónica.
- Se creará una interfaz gráfica de usuario que permita a personas poco familiarizadas con la temática realizar diversos estudios.
- Permitir al usuario que va a usar la interfaz la selección de los mejores electrodos, frecuencia y fase en la estimulación y detección de señales SSVEP para una mejor adaptación al usuario del BCI.
- Se proporcionará una exportación tanto de los archivos binarios generados por las grabaciones del equipo, como un fichero en formato JSON con información de la sesión del usuario para facilitar la integración de una base de datos.

La información se recoge de forma anónima para garantizar los derechos de privacidad.

- Se validará la aplicación en una prueba de registros de EEG con electrodos secos.

### 1.3 Estructura de la Memoria

En esta memoria se recoge una serie de capítulos que abarcarán los siguientes apartados:

- **Introducción:** Se detallan cuales son los motivos por los cuales se ha decidido hacer este trabajo y cuales son los objetivos a cumplir.
- **Estado del arte:** Proporciona una visión actualizada de las metodologías empleadas en la detección de señales SSVEPs en interfaces cerebro maquina (BCI) mediante la electroencefalografía (EEG). Unas tablas de resumen mostrarán una rápida comparativa de ventajas y desventajas entre ellas.
- **Diseño del sistema:** Muestra el diseño a seguir en el desarrollo de la plataforma y configuración del BCI. Esto engloba tanto componentes Hardware como Software, diseño de la plataforma y la interfaz gráfica. Intenta dar una visión simple del sistema para un mejor entendimiento y su posterior desarrollo.
- **Desarrollo del sistema** En este apartado se explicará las decisiones tomadas en la implementación y aquellos detalle que son relevantes. Además se explicarán aquellos problemas que han ido apareciendo a lo largo de todo el proceso.
- **Resultados:** Muestra los resultados de las pruebas realizadas para validar la plataforma. Además se provee de una guía paso a paso de cómo se debe hacer un experimento y los resultados asociados.
- **Conclusiones:** A través de los resultados obtenidos se sacarán las conclusiones oportunas. Finalmente habrá un apartado de trabajo futuro dónde se explicarán algunas ideas a implementar y combinar con lo desarrollado y que se plantean como un trabajo futuro.



## Capítulo 2

# Estado del Arte

### 2.1 Actividad cerebral humana

La evolución de la especie humana durante siglos ha dado como fruto un organismo capaz de pensar por sí mismo y de tener la consciencia de su propia existencia. Esta capacidad es gestionada por un complejo órgano llamado cerebro que entre otras cosas procesa los pensamientos y los movimientos tanto voluntarios como involuntarios, como son el respirar y el latir. Esto es debido a que se compone de millones de neuronas que se organizan formando una compleja red de regiones con distinta especialización. Algunas zonas se especializan en el procesamiento del lenguaje, otras en la visión y así hasta cubrir todas las capacidades de las que ha sido dotado y entrenado el cerebro.

Las neuronas de las que se compone el cerebro, reciben la información de otras neuronas a través de la dendritas. Dicha información es procesada generando así una nueva actividad que es enviada de nuevo a otras neuronas a través del axón (Kandel et al., 2012). Esta transmisión de información entre neuronas se realiza gracias a los neurotransmisores para las conexiones químicas y a las conexiones eléctricas que existen. De esta forma muchas neuronas se conectan entre sí formando una compleja red cuya dinámica puede cambiar en función de la propia actividad (Rabinovich et al., 2012).

La actividad que el cerebro produce es registrable y según los métodos de registrarla puede clasificarse en dos tipos fundamentales:

- **Respuesta hemodinámica**, la cual aparece cuando alguna zona del cerebro requiere gran cantidad de oxígeno y que puede ser registrada con técnicas de neuroimagen (Turkheimer et al., 2015).
- **Actividad electrofisiológica**, la comunicación entre las neuronas se realiza por transmisión tanto eléctrica como electroquímica y producen campos electromagnéticos registrables con distintas técnicas incluyendo las no invasivas (Nicolas-Alonso and Gomez-Gil, 2012).

#### 2.1.1 Tipos de registros de actividad del sistema nervioso

El sistema nervioso genera gran cantidad de señales, algunas de las cuales son más fáciles que otras a la hora de ser registradas y analizadas. En concreto, nos centraremos en aquellas señales que tienen que ver con el interés de realizar una acción por parte del usuario con un sistema de BCI, y las podemos clasificar en tres tipos fundamentales:

- **Evocadas**: También conocidas como Señales Evocadas Visualmente (VEP). Son generadas de forma inconsciente por la persona que recibe el estímulo y su origen se debe a una estimulación externa.

- **Espontáneas:** Al contrario que las Evocadas, éstas no necesitan un estímulo externo para ser generadas, pero sí cierto grado de voluntad por parte del usuario.
- **Híbridas:** Se define como la combinación de señales, que podrían complementarse en la misma intención o ser discriminatorias de información de estímulos o en voluntad del usuario.

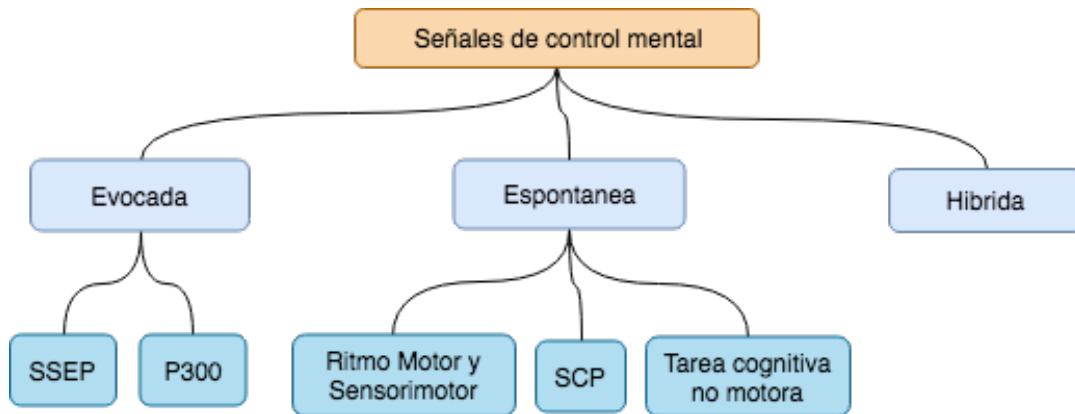


FIGURA 2.1: clasificación de señales cerebrales para su uso en BCIs.

- **Potenciales evocados de estado estacionario (SSEP):** Estas señales aparecen cuando se recibe de forma periódica un estímulo externo parpadeante, puede ser una imagen o una simple luz, y también un sonido repetitivo (Lim et al., 2013). El cerebro reacciona a este estímulo y genera una respuesta en la misma frecuencia de estimulación. El tipo de señal SSEP más conocido es la SSVEP, Potencial visual evocado de estado estacionario y que se refiere a recibir un estímulo visual parpadeante típicamente modulado a una frecuencia entre 5 y 90 Hz (Herrmann, 2001; Bin et al., 2009).
- **Potenciales visuales evocados (P300):** Al contrario que las SSVEP, éstas no necesitan que el estímulo sea constante por un periodo largo, se caracterizan por aparecer cuando un estímulo es poco frecuente o inesperado. Aparecen como combinación de señales en el cerebro y su duración es de los 300 milisegundos.
- **Ritmos motor y sensorimotor:** Estos ritmos son generados como consecuencias de la intención del usuario de realizar algún movimiento o su imaginación con las extremidades. Proceden de la corteza motora y los tipos de ritmos asociados son *Mu* y *Beta*, Tabla 2.2. Su amplitud puede ser controlada de dos formas:
  - **Condicionamiento operante:** a través de un largo periodo de entrenamiento el usuario es capaz de elegir su propia estrategia para cambiar la amplitud de los ritmos
  - **Imágenes motoras:** el usuario es capaz de imaginar que realiza un movimiento y en la corteza motora aparecen los ritmos asociados a este tipo de movimiento.



- **Potenciales corticales lentos (SCP)** Se caracterizan por tener una oscilación lenta por debajo de 1Hz cuya aparición pueden variar entre uno y varios segundos. Si los SCPs son positivos significa que la actividad neuronal disminuye, en cambio, si son negativos está relacionado con un aumento de actividad de las neuronas. El control de la oscilación puede ser realizado por personas sanas o con parálisis (Nicolas-Alonso and Gomez-Gil, 2012).
- **Tarea cognitiva no motora** Se refiere a tareas cognitivas que no requieren la imaginación o intención de mover alguna parte del cuerpo. Un claro ejemplo de este tipo de actividad sería la de realizar un conteo mental y de alguna forma imaginar o visualizar la operación (Ramadan and Vasilakos, 2017).

Tipo de señal	Principal característica	Tranf. bits/min	Entrenamiento necesario
VEP	Modulación de señal en la corteza visual	60-100	No
P300	Picos de voltaje con un estímulo infrecuente	20-25	No
SCP	Oscilación lenta de señales cerebrales	5-12	Si
R.Sensori motores	Modulaciones de actividades motoras	3-35	Si

TABLA 2.1: Tabla resumen de tipo de señales cerebrales y sus características (Ramadan and Vasilakos, 2017).

En la tabla anterior se muestran los tipos de señales ordenados de mayor a menor uso en BCI, siendo los SSVEPs y P300s los más habituales debido a alta tasa de transferencia de información por parte del usuario y de la no necesidad de un entrenamiento previo.

### 2.1.2 Monitorización e interpretación de señales del sistema nervioso

La actividad generada por el cerebro puede ser registrada mediante distintas técnicas y cada una de ellas tendrá sus propias características. Vamos a realizar una comparación centrándonos en dos aspectos importantes:

- **Resolución espacial:** permite determinar cómo de precisa es la localización de un evento de interés en un área cerebral en concreto.
- **Resolución temporal:** determina si el evento de interés se produce en un rango de tiempo determinado.

Una vez hemos explicado cómo vamos a comparar las técnicas procederemos a resumir las más usadas (Ramadan and Vasilakos, 2017):

- **Electroencefalografía (EEG):** Cuando un conjunto de neuronas generan actividad eléctrica, dicha actividad puede ser registrada por unos electrodos situados en la superficie externa de la cabeza. Estos electrodos se pueden clasificar en dos grupos: activos y pasivos, siendo los primeros los que requieren un amplificador externo; y los segundos los que llevan embebido en el propio electrodo. Además, según el tipo de colocación pueden ser clasificados en electrodos secos o con gel. Los que usan gel están mejor aislados de interferencias externas o de la mala colocación de los electrodos, pero es

poco práctico poner gel en cada electrodo cada vez que se use, sin contar con los problemas de higiene que puede causar. En cambio, los electrodos secos parecen ser los elegidos por su limpieza y portabilidad, a pesar de que no son tan precisos como los primeros (Ramadan and Vasilakos, 2017).

Según el rango de frecuencia en el que opera cada tipo de señal EEG y que generan distintos ritmos registrables, podemos clasificarlas en la siguiente tabla:

Tipo de ritmo	Rango de frecuencia	Actividad mental
<b>Delta</b>	1-3 Hz	Sueño profundo
<b>Theta</b>	4-7 Hz	Inconsciencia, meditación
<b>Alpha</b>	7-12 Hz	Relajación y concentración
<b>Mu</b>	8-13 Hz	Baja amplitud motoneuronas trabajando
<b>Beta</b>	12-30	Alerta, pensamiento y concentración
<b>Gamma</b>	>30 Hz	Uso de memoria a corto plazo

TABLA 2.2: Tabla resumen de tipos de ritmos cerebrales (Ramadan et al., 2015).

Por lo tanto para llevar a cabo el estudio sobre una determinada actividad cerebral se debe tener en cuenta el tipo de ritmo asociado.

Esta técnica tiene una resolución temporal muy buena pero la resolución espacial es baja, debido a que la señal tiene que atravesar el cráneo. Como consecuencia de ello las señales pueden mezclarse en los electrodos añadiendo ruido a la señal. Además las interferencias internas o externas al organismo también afectan a la calidad de las mediciones. A pesar de ello es la técnica más usada por ser barata, segura y portable, aunque también es la menos precisa por la cantidad de ruido que puede registrar (Ramadan et al., 2015).

- **Electrocorticografía (ECoG):**

Mide la actividad eléctrica del cerebro de una forma invasiva, esto significa que requiere de una operación quirúrgica para introducir los electrodos por debajo del cráneo y en la superficie del cerebro. Por las implicaciones que conlleva es una técnica poco usada y con muchas limitaciones, estando reservada a pacientes con enfermedades severas. En comparación con un EEG tradicional proporciona una mejor calidad, ya que registra las señales a través de capas blandas y los electrodos están mejor aislados del exterior (Nicolas-Alonso and Gomez-Gil, 2012).

- **Magnetoencefalografía (MEG):** Como hemos comentado anteriormente, la actividad cerebral produce corrientes eléctricas. Estas corrientes a su vez generan un campo magnético asociado y que también puede ser registrado. Al igual que el EEG, se registran de forma externa y en contacto con el cuero cabelludo, pero tienen mejor resolución espacial. Debido a que las ondas magnéticas se distorsionan más que las eléctricas, los electrodos deben ser aislados de interferencias electromagnéticas externas en una habitación debidamente protegida (Nicolas-Alonso and Gomez-Gil, 2012). Esta técnica es recomendada para tratar pacientes con tumores cerebrales o epilepsia ya que es capaz de localizar en el cerebro el origen del problema (Ramadan and Vasilakos, 2017).

- **Tomografía computerizada por emisión de fotón único (SPECT):** Esta técnica está diseñada para generar mapas en 3D de la actividad biológica. Para ello se usa una cámara de rayos gamma que detecta las emisiones de gamma que emite los radionúclidos, átomos radioactivos que por su inestabilidad emiten este tipo de rayos (Ramadan and Vasilakos, 2017).
- **Tomografía de emisión de positrones (PET):** Técnica similar a la SPECT, pero en este caso se le inyecta al paciente cierta cantidad de radionúclidos. Gracias a ello se tiene más información a la hora de generar los modelos 3D y poder registrar la actividad metabólica. Tiene un elevado coste que la hace poco usable (Ramadan and Vasilakos, 2017).
- **Imágenes de resonancia magnética funcional (fMRI):** Es una técnica no invasiva que detecta la cantidad de oxígeno que hay en la sangre durante la actividad cerebral. Tiene una alta resolución espacial, pero muy poca resolución temporal llegando incluso hasta los 2 segundos. Los problemas asociados a esta técnica son por ejemplo los movimientos de la cabeza mientras dura las pruebas, generando ruido y artefactos difíciles de corregir (Nicolas-Alonso and Gomez-Gil, 2012).
- **Infrarrojo cercano funcional (fNIR):** Tecnología que se basa en la reflexión de la luz infrarroja cuando se aplica sobre el cuero cabelludo. Es capaz de detectar zonas con alta cantidad de sangre y cambios en el nivel de oxígeno de la sangre. Esta técnica produce mapas 3D similares a los de fMRI, pero debido a que su resolución temporal es muy pobre es la menos elegida para realizar estudios (Nicolas-Alonso and Gomez-Gil, 2012).

Método de registro	Actividad medida	Resolución temporal	Resolución espacial	Invasivo	Portable
EEG	Eléctrica	$\approx 0.001$ s	$\approx 10$ mm	No	Si
ECoG	Eléctrica	$\approx 0.003$ s	$\approx 1$ mm	Si	Si
MEG	Magnética	$\approx 0.05$ s	$\approx 5$ mm	No	No
PET	Metabólica	$\approx 0.2$ s	$\approx 1$ mm	No	No
SPECT	Metabólica	$\approx 10$ s-30min	$\approx 1$ cm	No	No
fMRI	Metabólica	$\approx 1$ s	$\approx 1$ mm	No	No
fNIR	Metabólica	$\approx 1$ s	$\approx 2$ cm	No	Si

TABLA 2.3: Tabla resumen de técnicas de registro de actividad cerebral (Ramadan and Vasilakos, 2017).

## 2.2 Interfaces cerebro máquina (BCI)

### 2.2.1 Definición de BCI

La importancia de la interacción entre persona y computador (HCI, por sus siglas en inglés) viene determinada por su amplia aplicación en la vida humana, desde un ratón de ordenador hasta el ya conocido World Wide Web (Myers, 1998). Inicialmente tenía una usabilidad académica, pero pronto se vería expandido su uso al ámbito cotidiano.

Según (Wolpaw et al., 2002), un tipo de HCI serían las interfaces cerebro máquina. Se define como un sistema que integra distintas partes como son la electrónica y el Software y que se combinan para permitir la interacción de personas con problemas altas discapacidades.

Las interfaces cerebro máquina están compuestas por una serie de etapas o pasos que van transformando la información que genera el usuario, desde su origen en el cerebro hasta su aplicación o comando como paso final. En este trabajo nos centraremos en las interfaces cerebro máquina (BCI) basadas en electroencefalografía (EEG).

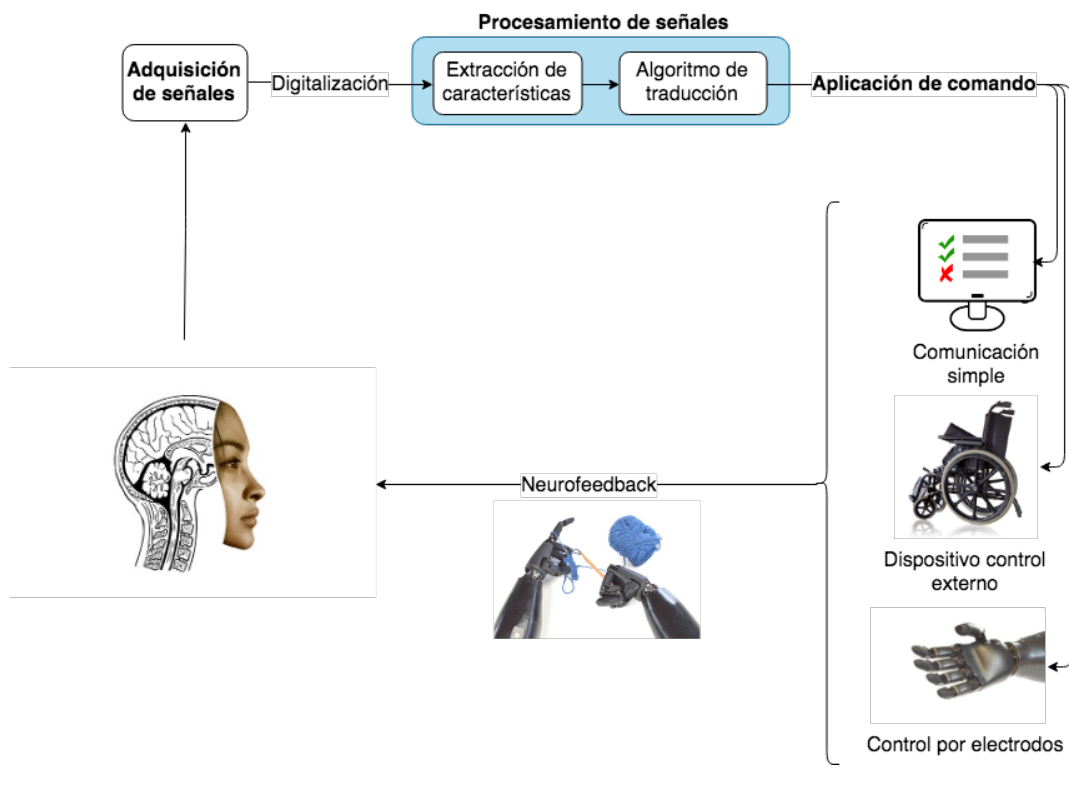


FIGURA 2.2: Imagen de un sistema BCI. (Combinación de imágenes sin copyright)

En la figura 2.2 podemos observar que un sistema BCI puede tener una alta complejidad. La aplicación va de la más sencilla, como es la elección de un comando en una interfaz gráfica, hasta la más compleja, como podrían ser un par de manos biónicas controladas por electrodos en un paciente que ha sufrido una doble amputación, y gracias a ello puede llevar una vida lo mas cercana a lo normal. Por tanto este tipo de sistemas tienen una repercusión de gran importancia para la vida cotidiana de las personas que tienen alguna deficiencia motora o para controlar otros sistemas (Wolpaw et al., 2002).

### 2.2.2 Tipos de BCI

Las interfaces cerebro máquina pueden clasificarse en muchos tipos y va a depender de si es una característica notable en su diseño, implementación o usabilidad. En este caso vamos a clasificarlos según el grado de dependencia e interacción que tiene con el usuario y el nivel de aplicación de los sistemas de registro de señales.

### 2.2.2.1 Dependientes vs. Independientes

La participación del usuario que usa el BCI juega un papel importante. Se considera que un BCI es dependiente si requiere alguna acción motora por parte del usuario. Por ejemplo, aquellas personas que han sufrido algún tipo de amputación y usen una prótesis que asista al movimiento. Por otro lado, los BCI independientes serán más útiles en personas con deficiencias motoras, puesto que no se requiere ninguna acción de movimiento por parte del usuario (Rodríguez Muñoz et al., 2018).

### 2.2.2.2 Invasivo vs. No Invasivo vs semi Invasivo

El grado en que un BCI es invasivo se refiere a cuanta profundidad deben ser colocados los electrodos para medir las señales del cerebro. En los No Invasivos los electrodos se colocan en la superficie del cuero cabelludo, son los menos precisos por la cantidad de capas que debe atravesar la señal pero los más fáciles de reemplazar y colocar. A estos les sigue los semi invasivos, los cuales requieren una operación quirúrgica y se compone de una pequeña malla de sensores situándola debajo del cráneo pero en la superficie del cerebro. Y por último los invasivos, que profundizan aun más en alguna región del interna del cerebro. Tanto los semi invasivos como los invasivos requieren de una operación quirúrgica. Tienen muy buena calidad de señal puesto que las señales no tiene que atravesar ninguna capa dura, pero son lo más complicados de llevar acabo por los riesgos que ello conlleva (Ramadan and Vasilakos, 2017).

### 2.2.2.3 Síncrono vs. Asíncrono

Nuevamente la participación del usuario tiene un efecto determinante, y en este caso se mide el grado de interactividad que tiene el usuario con el BCI. Se considera que es un BCI síncrono cuando requiere la participación del usuario en un tiempo determinado, de esta forma se tiene un control del avance a su voluntad. Por el contrario, se dará más libertad al usuario con los BCI asíncronos, dejando que realiza una acción sin restricciones de tiempo (Ramadan and Vasilakos, 2017).

## 2.3 Interfaces cerebro máquina (BCI) basados en EEG con señales SSVEPs

### 2.3.1 Motivo de elección de los SSVEPs

En este trabajo nos centraremos en las interfaces cerebro máquina (BCI) basadas en registros de electroencefalografía (EEG). Esta decisión viene motivada por varios motivos:

- **Alto ITR:** Como hemos visto en la tabla 2.1 los VEP son el tipo de señales que más información por minuto ofrecen y el más ampliamente usado. En comparación con los P300, su aparición en los registros desde la recepción del estímulo tiene un retraso muy bajo.
- **Portabilidad:** Son sistemas fácilmente ajustables a los usuarios debido que su colocación se realiza con una malla externa de sensores que debe colocarse de una determinada forma.

- **Trabajo previo:** este trabajo se postula como una continuación del trabajo previo realizado en el Grupo de Computación Biológica (GNB) (Rodríguez Muñoz et al., 2018; Rico Maldonado et al., 2018) y que será una forma de actualizar el desarrollo a nuevas tecnologías y facilitar el uso por parte de usuarios con conocimientos limitados.
- **Bajo coste:** A pesar de que puede contar con un equipamiento de altas prestaciones y coste, su implementación requiere de materiales de bajo coste y fácilmente reemplazables.

### 2.3.2 Etapas del sistema

En este trabajo nos centraremos en las interfaces cerebro máquina (BCI) basadas en registros de electroencefalografía (EEG). Como se adelantaba en la definición de un BCI, se descompone en una serie de etapas y que se puede englobar a grandes rasgos en: Adquisición de señal, Procesado de señal y Aplicación de comando. Además, en los sistemas de EEG basados en SSVEPs hay una etapa previa denominada Estimulación y que su diseño e implementación van a jugar un papel importante, pues suelen ser sistemas de fabricación poco estandarizados y con una amplia diversidad en precisión.

#### 2.3.2.1 Estimulación

Como ya hemos avanzado anteriormente, los sistemas BCI basados en EEG para detectar SSVEPs necesitan de un estímulo externo que parpadea de forma periódica. Este tipo de estímulo podría tener distinta fuentes y que según la bibliografía recogida se clasifican en:

- **Estimulación LED:** Son dispositivos electrónicos de bajo coste cuya implementación en la estimulación típicamente necesita de una circuitería básica controlada por algún ordenador o placa electrónica, como sería un Arduino, RaspberryPi o similar. Además se necesita una programación generalmente de bajo nivel, para controlar parámetros del LED como son la frecuencia, fase y duración del estímulo.
- **Estimulación por pantalla LCD** Típicamente es una pantalla de ordenador, tableta o móvil con una frecuencia propia que puede afectar a la selección de rango de frecuencias estimulable.

#### 2.3.2.2 Adquisición

Como sabemos las señales EEG se registran en la superficie del cuero cabelludo con unos electrodos preparados para ello. Los materiales de los que se componen han ido evolucionando a lo largo de los años para que sean por un lado conductores y por otro aislantes, puesto que son sensibles a interferencias externas. Podemos clasificarlos en:

- **Electrodos húmedos o de gel:** son aquellos que necesitan un gel conductor que favorece la conductividad evitando áreas de bajo contacto con la superficie y por consiguiente caída en el nivel de señal.
- **Electrodos secos:** Estos mejoran en adaptabilidad al no depender de ningún gel, pero en cambio pierden algo de precisión a los que utilizan gel conductor.

La posición de los electrodos es importante sobre todo si se van a comparar resultados, por eso existe el modelo 10-20 que es internacionalmente aceptado y ofrece una determinada distribución de los electrodos en una malla que cubre la cabeza.

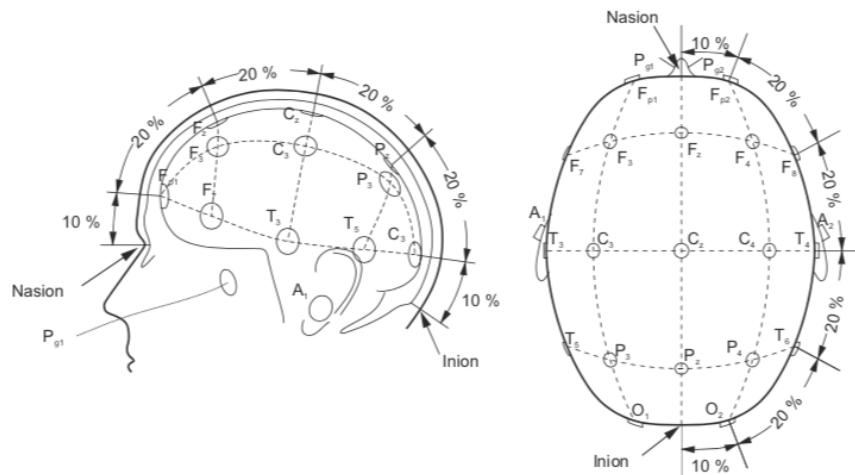


FIGURA 2.3: Imagen del modelo 10-20 de Electrodo EEG (Jasper, 1958).

Una vez se tiene el modelo, se optará por una malla de electrodos de algún fabricante o fabricación propia pero respetando el estándar.

La adquisición de señales se realiza a través dispositivos electrónicos de alta eficiencia que deben recoger la señal y pueden ser configurados para eliminar ruidos y artefactos de distinto tipo, esto quiere decir, que en la señal pueden haber ciertas interferencias eléctricas, electromagnéticas o de calor y que afectan en distinta medida la correcta recepción y clasificación de la señal. Para evitar esto, es común hacer un preprocesado de la señal usando filtros de diverso alcance.

### 2.3.2.3 Procesado

El procesado de información en las interfaces cerebro máquina juega un papel tan importante como el de la adquisición de la señal y se puede realizar en dos modalidades:

- **Procesado online.** El procesado de la información de las señales se realiza en un tiempo determinado y depende de la necesidad requerida del propio sistema. Este procesado adquiere gran relevancia en los BCIs.
- **Procesado offline** Este tipo de procesado está pensado más para un estudio posterior, sin requerimientos de tiempo. Aunque tanto el online como offline se pueden combinar dando al sistema una rápida respuesta y una posible corrección más tarde.

Otro aspecto importante es la neurofeedback que se pueda generar en estos sistemas. El usuario por su parte puede observar y aprender de la relación que existe entre estímulo-acción o causa-efecto, éste aprende de ello y lo puede usar para lograr un objetivo determinado. Por lo tanto, la eficiencia de estos sistema puede

verse aumentada con el uso del neurofeedback y aparece lo que se denomina como sistema de circuito cerrado (closed-loop system) (Martinez et al., 2007; Weiskopf et al., 2004). El grado de aprendizaje del usuario afectará en mayor o menor medida a la efectividad entre neurofeedback y el objetivo a alcanzar (Fernandez-Vargas et al., 2013a).

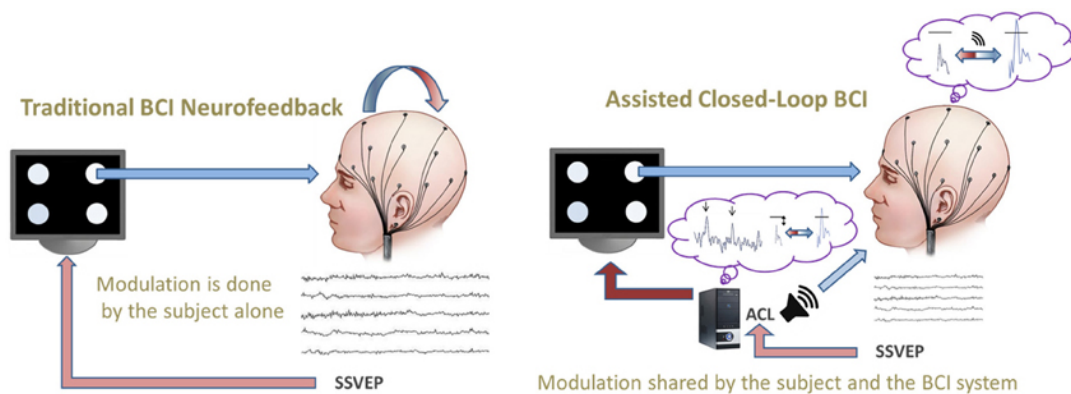


FIGURA 2.4: Imagen comparativa tipos de Neurofeedback en BCIs (Fernandez-Vargas et al., 2013a).

Una vez definidos los requerimientos de tiempos en cuanto a la entrega de información podemos decir que las metodologías implicadas en este apartado también mantienen la misma clasificación, pues el tiempo de computación necesario es un parámetro a tener en cuenta.

Para determinar qué información es relevante en los BCI hay que saber qué características necesitamos saber a priori, típicamente son características de la propia señal a analizar.

Para la extracción de características es necesario un preprocesado de la señal antes de realizar una interpretación de la información que contiene. Para ello es bueno aplicar los filtros necesarios, tanto temporales como espaciales, ya que la información está solapada en ambas dimensiones y se necesita discriminar y resaltar aquellas deseadas.

#### 2.3.2.4 Aplicación de comandos

La aparición de las interfaces cerebro máquina (BCI) no tendría sentido si no existiera una importante necesidad para ello. Muchas son las aplicaciones que existen y según la figura 2.2 del esquema del BCI, esta etapa tiene generalmente grandes expectativas por parte del usuario. El conocimiento que tiene el usuario sobre las interfaces cerebro máquina suele ser limitado, desconociendo todo el procesamiento que existe por debajo y su concepto sobre ello suele resumirse dos preguntas, cómo estimula y qué soy capaz de conseguir con ello.

Una vez el procesamiento de la señal ha dado como salida algunas de las frecuencias que se corresponden con las que se ha estimulado o se está estimulando, la aplicación del comando depende de la complejidad de esta parte del sistema. Los más comunes son:

- **Comunicación sencilla:** Suele ser alguna aplicación software como un teclado virtual en el que se muestra por pantalla un alfabeto mostrando cada letra con



parpadeo a una determinada frecuencia y fase. La aplicación más siempre es una configuración de LEDs distribuidos de una determinada forma.

- **Control dispositivo externo:** Aumenta en complejidad al incorporar elementos electrónicos y mecánicos como podrían ser una silla de ruedas motorizada y controlada por señales cerebrales.
- **Control por electrodos en los músculos:** Son los más complejos al añadir además del control de un dispositivo externo una red de electrodos que ayudan a realizar la actividad de forma más cercana a la humana. Por ejemplo un par de piernas con electrodos conectados al cerebro para activar el movimiento y sensores en músculos disponibles y útiles para controlar la fuerza del movimiento.

## 2.4 Estado del arte en detección y análisis de SSVEPs

A continuación vamos a resumir la revisión del estado del arte en cuanto a técnicas en el procesado de las señales SSVEPs, para ello podemos diferenciar tres ámbitos: filtros temporales, espaciales y combinados. Los más utilizados son:

- **Filtros temporales:**

- **Fast Fourier Transform (FFT):** Transformación de la señal del dominio del tiempo al dominio de las frecuencias, típicamente luego se aplica un filtrado de frecuencias (Rodríguez Muñoz et al., 2018).
- **Bandpass:** Es un filtro muy común y se basa en restringir el rango de las frecuencias en los que se descompone la señal. Para ello las amplitudes por encima y por debajo de dos límites de determinadas frecuencias se ponen a cero. Esto acota la información que se va a usar (Rodríguez Muñoz et al., 2018).
- **Butterworth:** Filtro cuya función es la de aplanar las frecuencias tanto como sea posible, posteriormente suele aplicarse el filtro Bandpass (Waytowich et al., 2018).
- **Spectrum Density Analysis (PSDA):** También conocido como densidad espectral, se refiere la transformación de la potencia de la señal en el espectro de frecuencias de la señal (Rodríguez Muñoz et al., 2018; Fernandez-Vargas et al., 2013a).

- **Filtros espaciales:**

- **Maximal-Phase-Locking Value:** Se usa para determinar la sincronización que existe entre dos señales por las diferencias de fase, éstas son la señal SSVEP y la de referencia que se genera como modelo a comparar (Lin et al., 2019).
- **Minimal-Distance Spatial Filter Banks:** Se basa en la obtención de la mínima distancia entre la señal SSVEP y la señal seno de referencia.

Para ello usa el coeficiente de Pearson y la distancia Euclídea (Lin et al., 2019),

- **Canonical Correlation Analysis (CCA)** Este enfoque mejora en precisión al PSDA y es ampliamente usado para la detección de SSVEPs. Busca la máxima relación lineal que hay entre dos conjuntos de datos, típicamente uno contiene las señales registradas y el otro las correspondientes señales de referencia de senos y cosenos a un determinada frecuencia. Las interferencias en la señal afectan seriamente a su rendimiento (Chen et al., 2018; Rodríguez Muñoz et al., 2018). Es la metodología que vamos a usar en nuestro trabajo y será explicada en detalle en el apartado de diseño y desarrollo.
- **Individual Training Data CCA** Metodología cuya aporte en comparación al CCA estándar es el de añadir en sus operaciones la información de entrenamiento previo con la misma frecuencia y fase, dando lugar a una mejora sustancial del CCA (Rodríguez Muñoz et al., 2018).
- **Filter Bank with CCA (FBCCA):** El procedimiento divide la señal en subbandas a las cuales se les aplica el filtros bandpass. Posteriormente a cada subbanda se le aplica CCA con su señal de referencia de senos y cosenos a una determinada frecuencia. Mejora en precisión al CCA (Rodríguez Muñoz et al., 2018).
- **Individual Training Data With Filter Bank CCA** Enfoque cuyo paso previo es el de aplicar un banco de filtros tanto a los datos que se van a analizar como los datos de entrenamiento previos, y así poder realizar una adaptación al usuario. Mejora en precisión al Individual Training Data CCA (Rodríguez Muñoz et al., 2018).
- **CCA Calibration data-based method** Metodología que mejora en precisión al CCA. Se basa en usar un banco de filtros y el CCA con datos de calibración individuales a los sujetos. El banco de filtros atenúa el efecto del ruido y los artefactos y los datos individuales mejoran la efectividad global de la metodología (Chen et al., 2018).
- **Spectrum and Phase Adaptive CCA (SPACCA)** Metodología que por un lado intenta reducir la distribución del espectro frecuencias entre usuarios mediante una adaptación del espectro; y por otro atenúa la distribución en los retrasos de los tiempos de respuesta del sujeto con respecto al estímulo usando un banco de señales de referencia con desplazamiento de fase. La metodología se puede adaptar a cualquier tipo de usuario y fuente del estímulo, y su eficiencia mejora la del CCA (Zhang et al., 2018).
- **Riemannian Geometry** Para solventar la variabilidad intra e inter sujeto utiliza datos adquiridos previamente para construir una geometría espacial determinada que ayuda a mejorar el clasificador, evitando así la etapa inicial de calibración cada vez que se usa el BCI. Este espacio

geométrico de Riemann estima la similitud, entre los datos a analizar y los previamente distribuidos geométricamente, de una base de datos multiusuario, es lo que se conoce como transferencia de aprendizaje, pues lo aprendido lo utiliza en posteriores análisis de señales del BCI. Para ello usa dos enfoques, Minimum Distance to Pooled Means (MDPM) y Minimum Distance to Weighted Means (MDWM) (Kalunga et al., 2018).

- **CNN-based network** Se propone una arquitectura de redes Neuronales Convolucionales cuyo diseño evoca la no necesidad de la etapa de selección ni de extracción de características. Para ello su framework convierte las señales en imágenes en escala de grises mediante la Fast Fourier Transformation (FFT), procesando los canales de forma independiente. Las redes neuronales no están pensadas para analizar datos en tiempo real, pero para solventar esto usan la técnica de ventana móvil. Para la validación de la metodología construyeron un sistema en tiempo real en el que el usuario controla un robot dándole éste un neurofeedback en forma de vídeo en tiempo real de los movimientos que realiza como respuesta a los comandos del BCI (Attia et al., 2018).
- **Common spatial pattern (CSP)** Procedimiento matemático que usa el filtro feedforward y que consisten en la comparación de la señal original con un subconjunto de copias de la original pero añadiendo distinto retraso a cada una. Posteriormente hay un reconocimiento temporal de patrones y clasificación de características. Este enfoque utiliza en las pruebas frecuencias de estimulación por encima de los 30 Hz, con el riesgo de provocar epilepsia al usuario por el solapamiento de tipos de señales, tabla 2.2. Otro aspecto destacable es el alto BCI Illiteracy detectado, el 60% de los usuarios no tuvo éxito en las pruebas (Chabuda et al., 2018).

- **Filtros combinados:**

- **Compact Convolutional Neural Network (Compact-CNN):** Enfoque con el cual no se necesita de una calibración para cada usuario ya que la extracción de características la realiza directamente de las señales SSVEPs, por lo tanto la información a priori no es necesaria. Esta técnica reduce la cantidad de datos de entrenamiento necesario, gracias a que aplica el Fast Fourier Transform (FFT) antes de proceder con la red neuronal. La extracción de características, es capaz de detectar no solo la frecuencia, sino también la fase y la amplitud. Mejora en precisión al CCA estándar (Waytowich et al., 2018).

### 2.4.1 Tabla resumen del estado del arte de BCIs que utilizan SSVEPs

En la tabla 2.4 tenemos la información referente a qué tipo de estímulo y qué métodos se han empleado en la revisión del estado del arte. Además en las columnas de Frecuencia, Fase y Armónicos se refiere a si la técnica ha optimizado su proceso para adaptar estos parámetros al usuario, por el cual se ha demostrado que mejora los resultados (Rodríguez Muñoz et al., 2018).

Con respecto a la tabla 2.5 podemos observar que en muchos casos se compara con el método estándar CCA, por lo que será la metodología para aplicar en este

trabajo. Algunos trabajos incluyen el ITR que mide el número de acciones que el usuario es capaz de hacer por minuto, típicamente es un juego deletreador con un abecedario como estímulo parpadeante dónde cada letra es un estímulo diferente y que mide cuantas veces se ha detectado la letra deseada en un tiempo determinado.

Autor	Estímulo	Frec.	Fase	Armo.	Métodos
(Lin et al., 2019)	LCD	No	Si	Si	Minimal-Distance (MD). Amplitude Maximal-Phase-locking value (MP).
(Waytowich et al., 2018)	LCD	Si	Si	Si	Compact-CNN. Teorema de convolución de las cuatro capas.
(Zhang et al., 2018)	LED LCD	No	Si	Si	Spectrum and Phase Adaptive CCA (SPACCA)
(Chen et al., 2018)	LCD	No	No	Si	CCA Calibration data-based method
(Kalunga et al., 2018)	LED	No	No	-	Minimum distance to pooled means (R-MDPM). Minimum distance to weighted means (R-MDWM)
(Attia et al., 2018)	LED Cámara	No	No	No	CNN-based network.
(Chabuda et al., 2018)	LCD LED	Si	Si	Si	Common spacial pattern (CSP) y Filtro Feedforward
(Rodríguez Muñoz et al., 2018)	LED	Si	Si	Si	PSD, max(SNR=S/B), FFT, CCA, FBCCA, IndTCCA, IndTBANKCCA

TABLA 2.4: Tabla resumen del Estado del Arte de técnicas usadas en detección de SSVEPs

## 2.5 Motivación del trabajo

De la revisión del estado del arte podemos deducir varias opiniones, por un lado existe un esfuerzo actual para lograr avances significativos en las interfaces cerebro máquina. Muestra de ello es la complejidad en las metodologías aplicadas y de reciente publicación, tabla 2.4

Por otro lado, y a pesar de los avances, no se consigue una efectividad del 100% ni con modelos matemáticos ni con aprendizaje automático, tabla 2.5. Como consecuencia podemos decir que no existe una metodología o técnica que garantice la robustez necesaria en los BCIs como para que su uso sea implantado en la sociedad. Por lo tanto, las interfaces cerebro máquina siguen siendo artefactos de laboratorio más que un producto usable en la vida cotidiana.

Hemos visto también que en algunos trabajos se usa una metodología o tecnología propia y que los resultados son extraídos a través de una implementación determinada en a configuración de los BCIs, tanto en componentes electrónicos como en Software desarrollado.

Basándome en los resultados de efectividad que se han obtenido en los trabajos, me surge la idea de si generar un punto de partida por el cual se estandarice el uso de una metodología en el BCI sea de gran ayuda. Este punto aporta cierta homogeneidad en las pruebas y detalles de implementación por el cual personas e instituciones podrían aportar su estudio. Buen ejemplo de ello sería que este punto

Autor	Acc %	ITR b/m	Aprendizaje Automático	Comentarios
(Lin et al., 2019)	92.6%	307	SVM	Sin optimizar: num. canales, frec. estímulo, num. estímulos y selección de electrodos
(Waytowich et al., 2018)	p-value	-	Deep Learning	Compact-CNN vs CCA: $t(9)=8.7$ , $p<0.0001$ ). Usa t-SNE en capas ocultas. BCI asíncrono.
(Zhang et al., 2018)	p-value	-	-	SPACCA vs CCA: Led(8.6%), PC(51.7%) y móvil(8.8%). Mejora significativa contra CCA $p\text{-value}=1.66e-6$ .
(Chen et al., 2018)	91.43% +-5.52	89.04 +-13.17	-	CCA Calibration data-based method vs CCA $p\text{-value}<0.001$
(Kalunga et al., 2018)	p-value	-	Bayesian classifier	Riemannian Geometry vs CCA $p\text{-value}=0.1$ .
(Attia et al., 2018)	94.67%	-	Deep Learning	Closed control loop. Feedback visual de robot con cámara.
(Chabuda et al., 2018)	96% +-4	47 +-5	Bayesian Classifier	Filtro Feedforward. Curva ROC. BCI-Illiteracy 60%. ITR incrementa con tareas simples. Frecuencias 30-39 Hz
(Rodríguez Muñoz et al., 2018)	Si	-	-	+13% Detección mejores frecuencias y mejores electrodos

TABLA 2.5: Tabla resumen del Estado del Arte y los resultados

común sea compartido por psicólogos, neurocientíficos y hospitales, generando todos ellos una visión particular y que la combinación de información pueden ser de gran utilidad.



## Capítulo 3

# Diseño del sistema

### 3.1 Objetivos en el diseño

El desarrollo de la plataforma requiere de una serie de compromisos para asegurar que se cumple la funcionalidad para la que fue creada y en un marco adecuado al usuario. Para ello en el diseño se deben tener en cuenta una serie de aspectos básicos.

#### 3.1.1 Simplicidad vs. Complejidad

Uno de los principales aspectos en el diseño es dimensionar la funcionalidad del sistema. Si se diseña con pocas funciones será más fácil de entender para el usuario. En cambio, si la funcionalidad aumenta la dificultad para el usuario crecerá de forma proporcional.

Conviene recordar que la plataforma está pensada para usuarios con conocimientos limitados en la materia, fomentando así su uso en distintos ámbitos biomédicos. Por lo tanto hay que buscar un equilibrio entre un desarrollo funcional o complejo. Una posible solución es la de diseñar la plataforma de tal forma que realice funciones complejas y mostrar una interfaz lo más simple y entendible posible.

#### 3.1.2 Usabilidad vs. Funcionalidad

En aspectos generales la usabilidad de un software juega un papel en contra de la propia funcionalidad de la aplicación. Una aplicación con mucha funcionalidad corre el riesgo de no tener la usabilidad necesaria para que el usuario siga las acciones del flujo de trabajo. Por lo que hay que tener especial cuidado y guiar al usuario en las acciones importantes de la aplicación.

Otro aspecto a tener en cuenta es que el incremento de funcionalidad hará crecer el número de potenciales errores en la propia aplicación. Como el diseño de este proyecto está pensado para realizar estudios científicos en el contexto de los interfaces cerebro-máquina, se optará por un diseño compacto y que permita al usuario una fácil adaptación a posibles cambios.

#### 3.1.3 Fácil de aprender vs. Fácil de usar

Siguiendo la filosofía de una aplicación compacta y aparentemente sencilla, el usuario podrá aprender de forma rápida qué es lo que puede hacer sin necesidad de tener un manual cerca. No todos los usuarios tienen interiorizado qué es lo que se puede hacer en este tipo de aplicaciones, ya que en principio no está pensando para que sea un producto comercial o estandarizado en la cultura digital. Por lo

tanto, hay una alta probabilidad de que para el usuario sea un modelo nuevo que deba de aprender y de la forma más rápida y fácil.

Una vez que el usuario ya ha aprendido a usar el modelo de trabajo, debe ser fácil de recordar todos los pasos que ha realizado o por lo menos recordar los principales y deducir los demás. Se plantea una plataforma que sea fácil de aprender, ya que con pocas acciones se puede realizar flujos completos de trabajo; y fácil de usar, el usuario será capaz de recordar las acciones principales y podrá retomar el uso de la aplicación en cualquier momento.

### **3.1.4 Modularidad y Reusabilidad**

Otro aspecto importante que se quiere conseguir en este trabajo es que el código desarrollado sea reusable y fácilmente ampliable. Para ello hay que determinar cuáles son las partes que pueden ser objeto de modificación en el futuro y cuáles no debido a la complejidad de su dependencia. Por ejemplo, en este trabajo vamos a usar el Análisis de Correlación Canónica como metodología principal para la detección de señales SSVEPs, pero la plataforma debe proporcionar la suficiente flexibilidad como para que futuros colaboradores puedan reusar esta plataforma y añadir otras metodologías. Para lograr dicha reusabilidad y facilitar nuevas mejoras, la plataforma debe ser diseñada y desarrollada pensando en la modularidad.

## **3.2 Diseño de la plataforma**

Siguiendo la filosofía de reusabilidad, este proyecto ha tomado como guía el trabajo realizado previamente por el Grupo de Neurocomputación Biológica (GNB). Por un lado tenemos el control de LEDs a través de un Arduino. Éste genera una estimulación con una onda sinusoidal que mitiga en cierta medida el cansancio del usuario en comparación con una onda cuadrada. Usaremos pues esta parte en el sistema BCI (Rico Maldonado et al., 2018). Por otro lado tenemos un sistema BCI montado que utiliza el Análisis de Correlación Canónica (CCA) y utiliza el sistema de estimulación antes mencionado. Dicho trabajo se ha desarrollado en diferentes lenguajes de programación, siendo el principal MATLAB (Rodríguez Muñoz et al., 2018). Este último ha servido para entender mejor la dinámica de un sistema BCI y a la vez para validar los resultados obtenidos de la propia plataforma que aquí se desarrolla. Para ello se ha utilizado la base de datos generada previamente por los mismos trabajos.

### **3.2.1 Lenguajes de programación**

En los trabajos citados anteriormente que hacen de guía propuesta, se hace uso de MATLAB. Este lenguaje de programación, además de necesitar un entorno de compilación propio, requiere de una licencia de pago que no todo el mundo puede permitirse. En caso de no disponer esta licencia no se puede desarrollar ni ejecutar ningún programa desarrollado en ese lenguaje, por lo tanto habría que buscar otras opciones para fomentar una máxima difusión de la plataforma.

Otro aspecto a considerar es la dispersión que existe con el número de scripts en diferentes lenguajes de programación. Por ello, la ejecución como un todo de un protocolo BCI basado en SSVEP con las herramientas desarrolladas anteriormente obliga a tener conocimientos técnicos en Linux, su shell BASH y otros lenguajes como MATLAB y C++, presentando una clara dependencia del uso del Software del BCI hacia el propio desarrollador.



La plataforma diseñada e implementada en este TFM unifica y amplía las herramientas desarrolladas anteriormente en un software que permite a cualquier usuario poder realizar diversos estudios con interfaces cerebro maquina basados en SSVEPs aislándolo de los típicos problemas del lenguajes de programación, variables dependientes, organización de carpetas, gestión de la documentación y problemas con el entorno de desarrollo. En definitiva, el usuario de la plataforma no necesita saber de Programación, Electrónica ni de la instalación de librerías en el ordenador, solo conocer el flujo de trabajo y qué puede conseguir con ello. Para llevar a cabo esta idea se ha optado por una visión de Open Source, reduciendo el número de lenguajes de programación a dos:

- Para trabajar con el equipo G-Tec, la propia compañía ha proporcionado un SDK con librerías en C++ para la interacción por Software con el dispositivo de amplificación de las señales de EEG. Se mantiene este pequeño código para agilizar el desarrollo.
- El lenguaje de programación principal elegido para el proyecto es Python, por tener una curva de aprendizaje baja y la fácil integración de librerías modernas. Si bien es cierto que en comparación con C++ no es lenguaje más rápido, pero en este proyecto no existe una restricción de tiempos demasiado estricta en cuanto ejecución de tareas.

### 3.2.2 Interfaz gráfica de usuario

En la figura 3.1 se presenta un diseño básico del que se partirá para su posterior implementación en la fase de Desarrollo:

Se propone una interfaz gráfica compuesta por 3 secciones:

- **Información de usuario y sesión:** En esta primera sección se guardarán detalles importantes del usuario tales como: si tiene gafas, el género, la edad y el alias de usuario, que se utiliza para evitar revelar el nombre real. La fecha de realización de las pruebas así como un código identificativo serán establecidos automáticamente. Es importante recalcar que la información de usuario se guarda de forma anónima. Posteriormente se puede guardar en un fichero encriptado la relación entre nombre de usuario y la verdadera identidad.
- **Sección de calibración:** Aquí entra en juego una serie de pruebas y procesamiento para determinar cuáles son las mejores frecuencias, fases y electrodos para el usuario a estudiar. Será explicado con detalle en el apartado de Diseño de fases. Una subsección mostrará como resultado los parámetros personalizados para el usuario.
- **Sección de Evaluación/Validación** Este paso es el que evalúa la compatibilidad de todos los parámetros. Para ello se lanzarán pruebas con todos los parámetros a la vez y cuyo resultado serán mostrados en un apartado. Permitirá al usuario seleccionar los parámetros obtenidos anteriormente y otros que el usuario de la interfaz decida en caso de que los primeros no sean favorables en las pruebas.

En la esquina superior derecha de la interfaz habrá una símbolo de una bombilla de color rojo, indicando al usuario de que la estimulación está en marcha. Además

FIGURA 3.1: Diseño interfaz gráfica de usuario.

habrá un botón que permitirá exportar a ficheros JSON la información registrada en cada sesión.

### 3.3 Diseño de etapas

El BCI que se utiliza en este trabajo tiene una serie de pasos que serán controlados a través de la interfaz gráfica anteriormente descrita. La ejecución de las pruebas implica que los pasos se hagan de una forma ordenada, y son los siguientes:

#### 3.3.1 Estimulación

1. **Configurar el estímulo:** Al realizar las pruebas de estimulación, primeramente se debe configurar el Arduino con los parámetros de frecuencia y fase necesarios para controlar el encendido y apagado de los LEDs de estimulación.

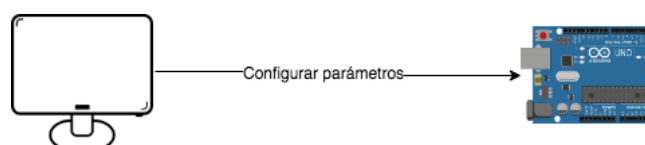


FIGURA 3.2: Configuración del estímulo.

2. **Lanzamiento del estímulo:** Una vez configurado el Arduino, lanzamos la orden de empezar a estimular, el amplificador empezará a grabar casi a la vez que activa la conexión con el Arduino, dejando 1 segundo de margen para tener una mejor sincronización de la señal, así sabremos que exactamente al segundo de la grabación empezará el estímulo (Rodríguez Muñoz et al., 2018). El Arduino a su vez encenderá los LEDs a una determinada frecuencia y fase anteriormente configurados. Y se apagará cuando el ordenador le diga que lo haga a través de un comando.

La interfaz permitirá la selección de un rango de frecuencias a estimular y la plataforma creará una lista de frecuencias entre dicho rango con una separación determinada. Posteriormente hará un barrido de exploración de la respuesta a dichas frecuencias de forma automática. Para empezar a ejecutar dicho barrido un botón permitirá el barrido automático, dejando al usuario de la interfaz gráfica más tiempo para comprobar que la prueba se está ejecutando correctamente.

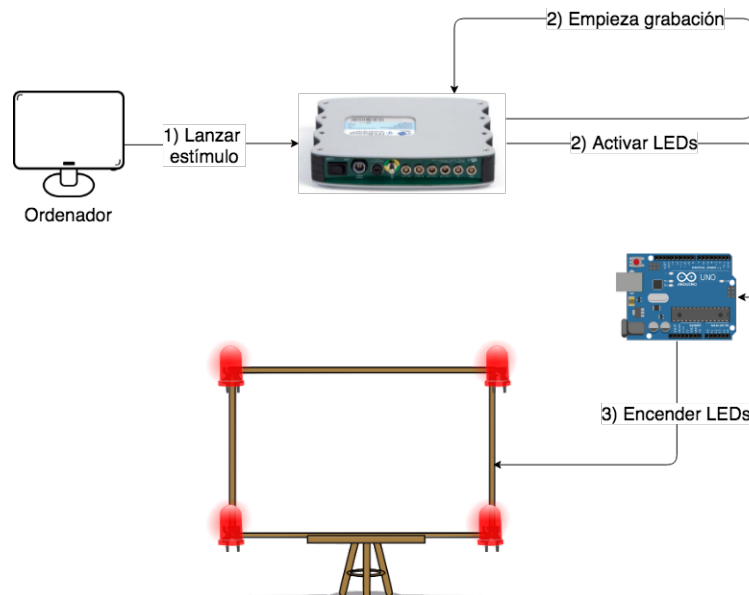


FIGURA 3.3: Lanzamiento del estímulo .(Combinación de imágenes sin copyright)

#### 3.3.1.1 Tipos de estimulación lumínica para evocar SSVEPs

La estimulación se realizará con LEDs debido a que por un lado el Grupo de Neurocomputación Biológica (GNB) ya contaba con el sistema montando, y por otro lado son componentes fáciles de reemplazar y su efectividad frente a monitores está demostrada ya que permite la utilización de cualquier frecuencia en el rango que genera SSVEPs (Fernandez-Vargas et al., 2013a).

Para determinar las frecuencias y fases óptimas en la adaptación del usuario, la estimulación se realizará con un solo LED y tantas veces como se determine en las pruebas, para poder tener suficientes datos como para hacer una media y evitar que las interferencias afecten a la correcta detección. En la fase de validación se estimularán los 4 LEDs a la vez y con los mejores parámetros, es decir, las cuatro mejores frecuencias, fases asociadas y electrodos para comprobar su compatibilidad.

### 3.3.1.2 Tipos de señales de estimulación

Hay dos formas de generar la señal de estimulación, generando desde el Arduino hacia los LEDS una señal de onda cuadrada o sinusoidal.

- **Onda cuadrada:** Estímulo normalmente molesto ya que pasa de una alta luminosidad a ninguna. El LED pasa de estar encendido a estar apagado de forma brusca, por lo que será molesto para el usuario y causaría posiblemente la falta de interés para la repetición de pruebas por parte el usuario (Fernandez-Vargas et al., 2013a).
- **Onda Sinusoidal:** Estímulo que sigue una onda que oscila de un valor positivo a negativo de forma suave. Es más cómodo para el usuario y sus ojos, por lo que potencialmente estará más dispuesto a realizar la prueba (Chen et al., 2015).

Según las pruebas realizadas en el trabajo (Rodríguez Muñoz et al., 2018) tanto la onda cuadrada como la sinusoidal tenían resultados parecidos en la detección de señales SSVEPs, pero los sujetos en pruebas preliminares manifestaron que la onda sinusoidal era menos molesta, por lo que se usará este tipo de señal de estimulación en el proyecto.

### 3.3.2 Adquisición de EEG

La adquisición de las señales es un proceso crítico que en los BCI se debe tener especial cuidado para proteger la señal de los electrodos de interferencias externas. Suelen ser sistemas cuya sensibilidad puede verse afectada incluso con la inestabilidad en el suministro eléctrico. Por ello hay que elegir bien el sistema según a las necesidades del proyecto.

Para la adquisición de señales EEG el Grupo de Neurocomputación Biológica (GNB) dispone dos sistemas:

- **Sistema EMOTIV EPOC:** Sistema de detección de señales que se compone de una malla de electrodos húmedos con un pequeño dispositivo en la zona occipital. Este dispositivo se comunica de forma inalámbrica con el ordenador para la transmisión de señales. En la localización de los electrodos se utiliza el modelo internacional 10-20 ya explicado. Dispone de 14 electrodos de posición fija y cuenta solo con dos electrodos en la zona occipital, la cual dificulta la detección de señales SSVEP (Rico Maldonado et al., 2018).
- **Sistema g-Tec:** Sistema de altas prestaciones que se compone de una malla de electrodos, un amplificador (g.USBamp) y un preamplificador (g.Sahara) con electrodos secos activos. El preamplificador tiene la función de proteger la débil señal que salen de los electrodos contra las interferencias externas. Dispone de 16 electrodos cuya posición es flexible y también sigue el modelo internacional 10-20. Este sistema es más preciso y más caro que el anterior.

Para el desarrollo de este trabajo se optará por el sistema de adquisición g-Tec por ser el más preciso y el que ofrece más flexibilidad a la hora de posicionar los electrodos para la detección de señales SSVEPs. Por otro lado, esta elección favorece la comodidad de los usuarios al no utilizar líquidos o gel conductor.

### 3.3.3 Procesado

La fase de estimulación y procesado se lanzarán desde el mismo comando de la interfaz gráfica, proporcionando simplicidad en el flujo de trabajo. Una vez la interfaz ha terminado con la estimulación buscará los registros grabados para un procesado de la información y hará un análisis online, mostrando al usuario de la interfaz la frecuencia detectada para comprobar que todo va bien. Al finalizar el número de repeticiones configurado en la interfaz, se hará un análisis offline con todas las grabaciones del mismo tipo de estímulo.

#### 3.3.3.1 Elección de metodología de detección de SSVEPs

En la revisión del estado del arte sobre metodologías en la detección de señales SSVEPs vimos que en muchos trabajos se comparaba el nuevo aporte con el ya estandarizado método del Análisis de la Correlación Canónica (CCA), además vimos que los resultados mejoraban pero no llegaban al ideal. En este trabajo el objetivo será usar esta metodología para permitir el desarrollo de un BCI completo controlado únicamente por una interfaz gráfica. Esta metodología será explicada en detalle más abajo.

En este paso se deben determinar cuáles son los parámetros que mejor se ajustan al usuario, los cuales serían: las frecuencias y fases asociadas y los electrodos. Para ello se necesitan los datos previamente registrados como consecuencia de la etapa de estimulación. Posteriormente se lleva a cabo un análisis de las señales y que se compone de varias metodologías:

- **Análisis de la Correlación Canónica (CCA)**

Se define como una metodología multivariable que tiene gran acogida en el campo de la estadística. Busca la relación lineal que pueda existir entre dos conjuntos de datos, estos son por un lado el conjunto de datos de las señales EEG registradas y por otro el de los datos con las señales de referencia, como serían las señales sinusoidales utilizadas para generar el estímulo visual del BCI. El método buscará las posibles combinaciones lineales entre ambos grupos y buscará la mejor correlación (ver Figura 3.4).

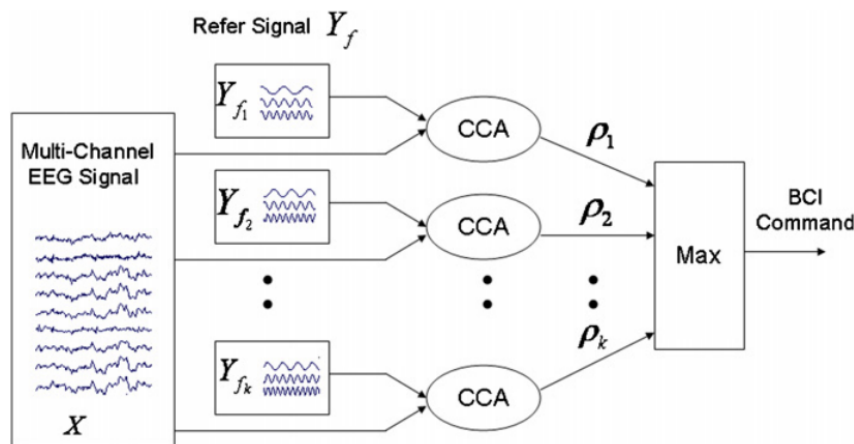


FIGURA 3.4: Imagen del Análisis de la Correlación Canónica (CCA) (Bin et al., 2009).

Siendo  $X$  el conjunto de señales EEG registradas por los electrodos y el conjunto  $Y$  las señales de referencia, tanto  $X$  como  $Y$  tienen la misma longitud. Se buscan las combinaciones lineales asociadas ( $x = X^T W_X$ ) e ( $y = Y^T W_Y$ ) respectivamente, siendo  $W_X$  y  $W_Y$  los vectores de pesos de las ecuaciones lineales. Se quiere conseguir la máxima relación que pueda existir entre ambos conjuntos.

$$\max_{W_X, W_Y} \rho(x, y) = \frac{E [W_X^T X Y^T W_Y]}{\sqrt{E [W_X^T X X^T W_X] E [W_Y^T Y Y^T W_Y]}}$$

Como comentábamos el conjunto  $Y$  se compone de las señales de referencia a una determinada frecuencia  $fn$  utilizada en la estimulación:

$$Y_f = \begin{bmatrix} \sin(2\pi fn) \\ \cos(2\pi fn) \\ \vdots \\ \sin(2\pi N_h fn) \\ \cos(2\pi N_h fn) \end{bmatrix}, n = \frac{1}{f_s}, \frac{2}{f_s}, \frac{3}{f_s}, \dots, \frac{N_s}{f_s}$$

Donde  $f_s$  es la frecuencia de muestreo,  $N_s$  es el número de puntos de la señal y  $N_f$  es el número de armónicos en los que se descompone la señal mediante Fourier.

- **Coefficiente de correlación**

Debido a que vamos a utilizar el método CCA elegiremos para el desarrollo la librería estándar de *sklearn*. El CCA de esta librería tiene como entrada dos conjuntos de datos para ser analizados y como salida dos conjuntos que representan los pesos en la combinación lineal de la entrada de la máxima correlación. Hay un componente final de esta estrategia de detección de los SSVEPs ilustrada en la figura 3.4 no incluido por defecto en la librería, por lo que necesitamos una metodología complementaria que correlacione estos dos nuevos conjuntos. Para ello vamos a elegir la librería *scipy* que es ampliamente conocida en el mundo estadístico y que ofrece dos tipos de correlación:

- **Coefficiente de Spearman:** se usa para buscar la correlación entre dos conjuntos con una relación monotónica, es decir, dos variables tienden a cambiar a la vez pero no al mismo ritmo, por lo que no puede ser lineal.
- **Coefficiente de Pearson:** busca la correlación que hay entre conjuntos con una relación lineal, es decir, dos variables cambian a un ritmo constante. Por ello vamos a elegir esta metodología para completar el análisis de la correlación canónica.

- **Relación de coeficientes de correlación (RCC)**

Una vez que tenemos una magnitud que representa el resultado del análisis de la correlación canónica entre dos conjuntos, tendremos que establecer si la frecuencia que se buscaba está o no en los registros de señales de EEG y por tanto coincide con el estímulo donde el usuario ha fijado su atención visual. Una vez terminado el análisis en todas las frecuencias, disponemos de una lista de coeficientes de correlaciones, y podremos hacer un ratio para medir cuál el máximo coeficiente asociado a una frecuencia y cuál es su relación con los demás. Usaremos esta fórmula:

$$\rho_{RCC} = \frac{\rho_{max}}{\rho_{no\_max}}$$

Donde  $\rho_{max}$  es el máximo coeficiente de Pearson que se asocia a la frecuencia detectada y  $\rho_{no\_max}$  es la media de todos aquellos coeficientes restantes que no son el máximo.

- **Metodología conjunta**

Por cada conjunto de grabaciones o trial se analizan todos los canales de forma conjunta e iterativa en el rango de frecuencias fijado por el usuario en la interfaz gráfica, y que corresponden a la estimulación. En cada iteración se obtiene una correlación máxima gracias a la aplicación del CCA. Por ultimo aplicamos el ratio RCC para comparar las correlaciones entre distintas sesiones.

Resumiendo, cada vez que se realiza el análisis de CCA tanto online como offline, se está identificando la frecuencia del estímulo en el que el sujeto pone su atención visual detectada con el coeficiente de Pearson y calculando el RCC asociado a tal detección. Ahora solo queda ordenar las frecuencias por valor del RCC y tendremos un *ranking* de las mejores frecuencias para ese sujeto.

### 3.3.3.2 Detección de mejores frecuencias

Como se ha comentado anteriormente, la interfaz permitirá la selección de las mejores frecuencias para cada sujeto. La plataforma estimulará de forma automática con la lista de frecuencias especificada recorriéndola de un extremo al otro con el incremento de 1 Hz y hará un análisis rápido CCA, proporcionando además un feedback al usuario de la interfaz del resultado de dicho barrido. Cada barrido o recorrido de lista de frecuencias se considera un trial y el número de trials es configurable en la interfaz. Una vez se exceda este número la plataforma hará un análisis offline con todos los ficheros generados dando una lista ordenada de frecuencias en función del ratio y el número de aciertos en la detección de la frecuencia mediante el Análisis de la Correlación Canónica.

### 3.3.3.3 Selección de mejores fases

Para la selección de la mejor fase de encendido de cada LED, se elige de la lista de frecuencias del paso anterior aquella que tiene el valor del ratio más alto. Ahora se crea una nueva lista que relaciona la misma frecuencia con distintas fases para posteriormente realizar el barrido de estímulos. Esta lista de fases va con un incremento de  $0.5 * \pi$  y de un rango de  $0 * \pi$  hasta  $1.5 * \pi$ . Se podría realizar una exploración con mayor resolución de las fases de todas las frecuencias de la lista original, pero entonces el periodo de pruebas se alargaría considerablemente. No olvidemos que se necesitan a personas para hacer este tipo de pruebas y que tienen que poner la atención visual en un estímulo durante el tiempo que dure.

### 3.3.3.4 Selección de mejores electrodos

De la detección de las mejores frecuencias se obtiene como resultado una lista de frecuencias y su ratio de correlación RCC asociado. Para la detección de los mejores electrodos usaremos los datos de registros asociados a la detección de la mejor frecuencia, teniendo así un mejor resultado en los datos para la clasificación. Una

vez tenemos los datos de registro procederemos de la siguiente forma: para cada electrodo eliminamos su señal en los datos de registro y hacemos el análisis de señal, si el ratio calculado mejora al de referencia, la cual se calcula con todos los electrodos, significa que ese electrodo en concreto añade ruido. En caso contrario, se considera un electrodo que aporta una información valiosa para el procesamiento de la señal. Por lo tanto se irá realizando la misma operación con todos los electrodos. Finalmente, tendremos una lista de electrodos con su correlación RCC de tal forma que están ordenados de menor a mayor RCC, recordando que un menor ratio significa que si ese electrodo se quita el coeficiente baja, por lo tanto es más importante tener un RCC bajo.

### 3.3.4 Aplicación de comandos y evaluación del rendimiento

La aplicación del comando en este sistema BCI se basará en proporcionar un sistema de puntuación para determinar la eficiencia del BCI y de los mejores parámetros con el usuario.

Debemos de tener en cuenta que los parámetros que mejor se ajustan al usuario han sido calculados de forma independiente, esto quiere decir que pueden haber algún tipo de incompatibilidades entre ellos cuando se configuran y se ejecutan a la vez en el mismo BCI.

La plataforma validará la compatibilidad que puedan tener dichos parámetros mediante la estimulación conjunta y el análisis de las señales. Como hemos explicado anteriormente, la plataforma transformará cada barrido de estimulaciones de frecuencias en una lista de frecuencias detectadas por la metodología especificada. Ahora realizando un proceso iterativo con las frecuencias de estimulación y las detectadas en el registro de EEG, sabremos cuáles han coincidido y cuántas veces se ha acertado, es decir por cada frecuencia tendremos un número de aciertos.

Además existirá un sistema de puntuación que permitirá saber el porcentaje de acierto total comparado con el máximo teórico. La puntuación mide cuántos aciertos han habido y lo divide por el máximo teórico, es decir, por el número total de trials que se han realizado para la estimulación de esa frecuencia. Si se han realizado dos barridos de rangos de frecuencias, el número dos será el máximo teórico.

$$NumAciertos_{f_{total}} = \frac{NumAciertos_f}{NumAciertos_{f_{max}}}$$

Posteriormente se pasa a porcentaje y tendremos un valor cualitativo de nuestra detección de frecuencias y con una configuración de parámetros determinada, y por consiguiente, se puede comparar con otras detecciones y con otra configuración para determinar la mejor adaptabilidad al usuario en las interfaces cerebro máquina.



## Capítulo 4

# Desarrollo del sistema

En este capítulo se van a exponer en detalle todos aquellos elementos del sistema que hayan sido creados o utilizados para construir la interfaz cerebro máquina

### 4.1 Interfaz gráfica de usuario (GUI)

Primero se explicará cuáles han sido las tecnologías implicadas y el motivo de su elección, luego se detallará el programa usado para facilitar el desarrollo y finalmente la propia interfaz gráfica.

#### 4.1.1 Metodología en el desarrollo

El desarrollo de este proyecto se ha realizado en su mayoría en el mismo lenguaje de programación, Python. En principio este lenguaje carece por defecto de librerías destinadas al entorno gráfico. Para solventar esto hemos seleccionado QT como tecnología implicada en este proceso.

Para crear la interfaz se necesita algún editor de texto plano, pero como realmente es bastante tedioso trabajar así optamos por un IDE o entorno de desarrollo llamado QTDesigner, el cual permite arrastrar y soltar de una paleta de componentes visuales y que facilitan en gran medida la localización final de los componentes en la interfaz de la plataforma.

Una vez se han creado los elementos y de la ventana se guarda en formato \*.ui. Para facilitar la interacción entre la lógica de la plataforma con los objetos de la ventana se ha decidido convertir el fichero de la interfaz a uno con formato \*.py. De esta forma, se pueden llamar los objetos de la ventana como si fueran objetos de programación, con sus propiedades y métodos. Una forma de conversión es la siguiente.

- `pyuic5 screen_main.ui > screen_main.py`

Hay que tener cuidado en este paso, pues el compilador suele añadir algún `_rc` al final del nombre interno de la clase interfaz y que a la hora de importarlo en la plataforma usa el mismo nombre.

#### 4.1.2 Interfaz gráfica

Como se adelantaba en el capítulo de diseño se ha optado por un diseño compacto, fácil de seguir e intuitivo. Dispone únicamente de una ventana principal dividida en sección que serán los pasos a seguir para utilizar el BCI de forma correcta.

The screenshot shows the SSVEPAdapt v1.0 GUI. The title bar reads 'SSVEPAdapt v1.0 by GNB-UAM | User adaptation for SSVEP-BCIs'. The interface is divided into several sections:

- User and Session Information:** Contains fields for Username, Session name, Session date (31/07/2019 17:38), Gender (Male), Glasses (checkbox), Age (18), Session cod, and Comments. A 'Save' button is at the bottom right.
- Step 1: Configuration:** Includes a 'Map' dropdown (Standard EEG map 10-20) and a 'USB port' text field (/dev/ttyACM0).
- Step 2: Calibration - Best Frequencies and Electrodes:** Features 'From' (8.00 Hz), 'To' (15.00 Hz), and 'Trials' (3) input fields with a right arrow button.
- Step 3: Calibration - Best Phases:** Features 'Freq.' (8.00), 'From' (0.00), 'To' (1.50), and 'Trials' (1) input fields with a right arrow button.
- Step 4: Validation:** Includes an 'Application' dropdown (Online Test Application), 'Trials' (3) input field, and three radio button groups for 'Using best: Electrodes', 'Frequencies', and 'Phases', each with an 'Other' option and a text field.
- Results:** Located on the right, it contains three sections: 'Best Elect.' with an empty text field; 'Best Freq.' with an empty text field and '0 trials' / 'Clear' buttons; 'Best Phase' with an empty text field and '0 trials' / 'Clear' buttons; 'Detected Frequencies' with an empty text field and '0 trials' / 'Clear' buttons; and 'Final Score' with an 'Export to JSON' button.

FIGURA 4.1: Interfaz gráfica de usuario (GUI)

#### 4.1.3 Secciones en la interfaz gráfica de usuario

- Sección de información de usuario y sesión:** La primera sección tenemos los campos relacionados con el usuario y la sesión. Nombre de usuario, género, si usa gafas, la edad que tiene y zona para escribir algún comentario relevante para las pruebas. Con respecto a la sesión tenemos el nombre de la sesión, y la fecha en la que se realiza, esta última puede ser modificable. Hay un campo no modificable y cuyo contenido es el código de sesión, que mezclará el nombre y de la sesión y la fecha y servirá para generar un identificador único, pensando más bien para la integración de esta información en una base de datos externa. El botón de *Save*, generará la estructura de carpetas necesaria para almacenar todos los ficheros generados en la sesión. Tanto ficheros \*.bin por parte del SDK de g-Tec y renombrados adecuadamente, como un archivo JSON con los parámetros de la sesión. Para esto último hay un botón en la parte inferior de la ventana.
- Sección de configuración:** Esta parte está pensada para que se puedan seleccionar el modelo de localización de los electrodos. Además se informa del puerto USB al que se conecta el Arduino, encargado encender y apagar los LEDs.

- **Sección de Calibración Frecuencias y Electrodo:** Únicamente hay que seleccionar cuál es el rango de frecuencias a estimular y cuántas veces se hace el barrido. Hay un botón con una flecha para ejecutar la acción
- **Sección de Calibración Fases:** La frecuencia será obtenida automáticamente al acabar el paso anterior, pero permitirá su modificación. Además será posible seleccionar el rango de fases con el que se hará el barrido, y el número de veces que se repite. Un botón permitirá la acción.
- **Sección de Validación** Permite seleccionar el tipo de validación que se quiera hacer y el número de veces a repetir. Los parámetros que se ajustan al usuario serán seleccionados por defecto pero se permitirá cambiarlos si los resultados no son buenos
- **Sección de resultados** Muestra en cada sección cuál es el resultado y el números de repeticiones o trials, que se han llevado a cabo. Existe un botón de Clear asociado a cada paso de ejecución, de esta forma si algo no ha salido como se esperaba, se puede repetir solo esta parte y no todo de nuevo.
- **Sección de iconos:** En esta versión de la plataforma solo se contempla la estimulación LED, por lo cual el icono asociado estará habilitado, mientras que el de estimulación por pantalla estará de deshabilitado. Hay un icono en forma de luz roja y que se mostrará de un color intenso cuando el sistema esté trabajando en algunas de las partes. Esto indicará al usuario que debe esperar los resultados.

## 4.2 Configuración del BCI

### 4.2.1 Estructura de la plataforma

Uno de los aspectos que se ha intentado cuidar es el número de ficheros de la propia plataforma y que se compone de los siguientes:

- **screen\_main.py** Contiene solo la estructura de la interfaz gráfica, así como las propiedades típicas de localización, colores, y aspectos similares, pero sin funcionalidad.
- **resources\_rc.py** Fichero que contiene los recursos multimedia, como son por ejemplo el icono del Grupo de Computación Biológica (GNB) y los iconos del tipo de estimulación y el del LED de aviso de grabación.
- **main.py** Contiene el core de la plataforma en cuanto a tratamiento de ficheros, metodologías y filtros y análisis de las señales. El desarrollo se ha separado de la interfaz gráfica para permitir hacer pruebas de validación y análisis de datos de una forma rápida y flexible, está pensada para el usuario desarrollador.
- **launcher.py** Es el encargado de configurar la interfaz gráfica, tanto de propiedades como eventos, y contiene la funcionalidad necesaria para abordar la interacción entre el usuario y el core de la plataforma.

Como se puede ver, el desarrollo de la plataforma está pensado para que sea reutilizable por futuros trabajos permitiendo que los cambios afecten lo menos posible al resto de módulos.

### 4.2.2 Estructura de ficheros

La organización de los ficheros que genera la plataforma debe ser intuitiva, para que otras personas puedan empaquetar los ficheros en un fichero ZIP y sea portado a otro sistema.

- **Carpeta principal:** Contiene los ficheros de la plataforma antes nombrados y es el punto de partida de otras subcarpetas.
- **Subcarpeta gUSBamp\_files** Contiene todo lo necesario para que se pueda interactuar con el SDK de g-Tec. Necesita además de la previa instalación de un controlador que diga al Sistema operativo cómo debe comunicarse. Esta instalación está documentada en la propia documentación del dispositivo.
- **Subcarpeta records** Contiene todas las grabaciones agrupados por código de sesión. Cada usuario puede tener multiples códigos y por lo tanto carpetas.
- **Subcarpeta cod\_session** Subcarpeta dentro de *records* y cuyo nombre se refiere al código de la sesión que se ha grabado, incluyendo el nombre de la sesión si se proporciona y la fecha, esta última es obligatorio. Contiene un fichero JSON como exportación de parámetros de la sesión y que contiene información como: mejores frecuencias, fases y electrodos, número de trials en cada sección e información de usuario. Además todos los ficheros generados como consecuencia de la estimulación y registro y cuyos nombres van a depender de la fase del flujo de trabajo.

Las nomenclaturas con las que se generan los ficheros son:

- **record\_ $T_i$ \_freq\_ $F_i$ .bin:** Genera los ficheros en la etapa de estimulación y detección de mejores frecuencias.
- **record\_ $T_i$ \_online\_ $F_i$ .bin:** Genera los ficheros en la etapa de estimulación y detección en la fase de validación final.

Donde  $T_i$  es el número incrementativo del trial y  $F_i$  la frecuencia con la que se ha estimulado en el proceso iterativo que recorre la lista de estimulación.

### 4.2.3 Sistema de estimulación

Siguiendo el diseño del sistema estimulación de la interfaz cerebro máquina de la figura 3.3, el sistema se compone de un ordenador que gestiona los comandos, un amplificador, una placa Arduino y una estructura con 4 LEDs. Por lo tanto se ha mantenido el sistema original y que ha sido desarrollado en conjunto por (Rico Maldonado et al., 2018) y (Rodríguez Muñoz et al., 2018).

#### 4.2.3.1 LEDs

Estructura semi rígida que se compone de 4 LEDs conectados a una protoboard y esta a su vez al Arduino. Antes de cada estimulación se configura la frecuencia y fase de cada LED y posteriormente se activan. Los números escritos cerca de cada LED indicará la secuencia a la que el usuario debe prestar atención visual.

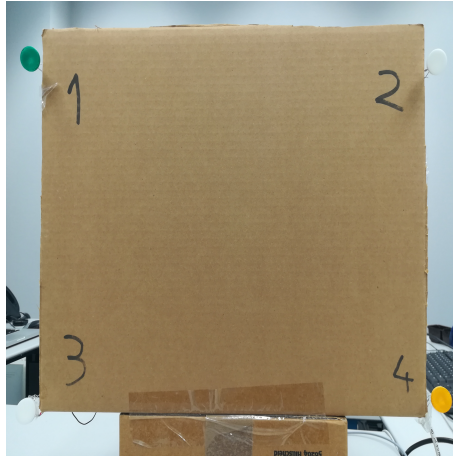


FIGURA 4.2: Montaje de LEDS para la estimulación visual.

#### 4.2.3.2 Arduino UNO

Se ha optado por reutilizar el Arduino el cual ya estaba configurado gracias al trabajo de (Rico Maldonado et al., 2018) y (Rodríguez Muñoz et al., 2018). Es un pequeño dispositivo que tiene lo que se necesita en este proyecto. Por un lado una conexión USB para ser programado y tener un terminal de escucha; y por otro una conexión interna de PWD, con la cual poder generar una onda sinusoidal para la estimulación de los LEDs y que se ha explicado anteriormente.

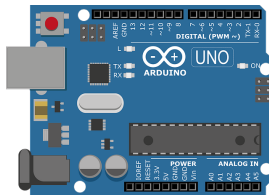


FIGURA 4.3: Imagen de Arduino Uno

El lenguaje de programación con el que se programa es C++ y cuando se le envían los comandos de configuración se hace a través del puerto USB y de la interfaz gráfica. Para ello se ha desarrollado un método en la plataforma que centraliza toda comunicación entre el PC y el Arduino.

#### 4.2.3.3 Sistema de estimulación integrado

Como se puede ver la figura 4.4 hay una protoboard que conecta los LEDs con el Arduino y este con el sistema g.USBamp a través de una entrada digital. En cada etapa del flujo de trabajo del experimento, la plataforma Software se encarga de enviar la señal que controla la placa Arduino para generar un estímulo visual con la frecuencia y fases deseados.

#### 4.2.4 Sistema de grabación

El sistema de grabación se compone de una malla de electrodos, un amplificador (g.USBamp) y un preamplificador (g.Sahara) para los electrodos secos activos.

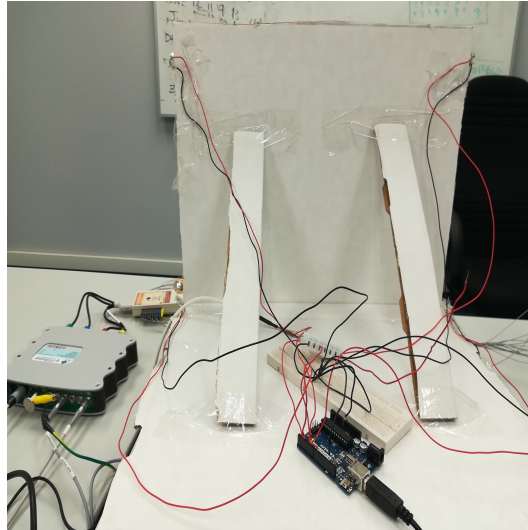


FIGURA 4.4: g.USBamp conectado al Arduino, al ordenador y a la tierra

#### 4.2.4.1 Electrodo seco activo

Estos electrodos tienen 8 pines cubiertos de una aleación de oro y tienen la suficiente longitud como atravesar la capa de pelos y llegar al cuero cabelludo en la mayoría de las personas. Existen otros con longitud más larga, pero solo serían útiles con pelo voluminoso, ya que de lo contrario podrían hacer demasiada presión en la piel y ser molestos o doblarse no quedar bien posicionados y ser poco efectivos. Este tipo de electrodos tiene un rango de sensibilidad de entre 0.1 y 40 Hz por lo que cubren el rango de señales SSVEPs.



(A) electrodo seco

(B) electrodo + conexión

(C) electrodos de referencia

FIGURA 4.5: Imágenes de los tipos de electrodos usados

#### 4.2.4.2 Malla de electrodos

La posición de los electrodos es importante, así como seguir el modelo internacional 10-20 y utilizar la configuración para la detección de señales SSVEPs.

Para conseguir este posicionamiento estándar necesitamos de una malla fabricada de tal forma que permite distintas configuraciones; y con materiales flexibles, ya que cada cabeza tiene una forma y dimensión determinada. Para ello necesitamos la malla de g.GAMMASys la cual ofrece distintas medidas y cuyo materiales son flexibles y transpirables.

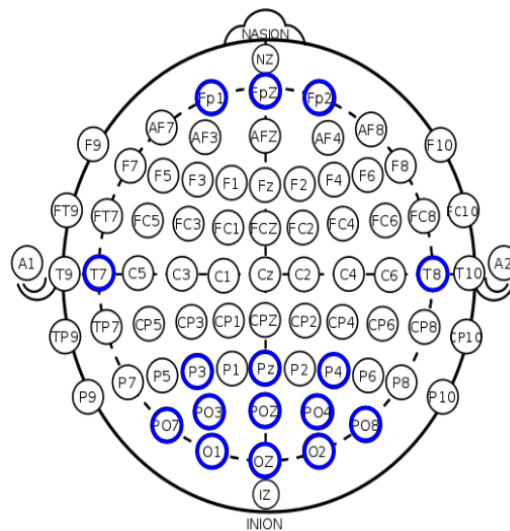


FIGURA 4.6: Localización de electrodos para la detección de SSVEPs. (Combinación de imágenes sin copyright)

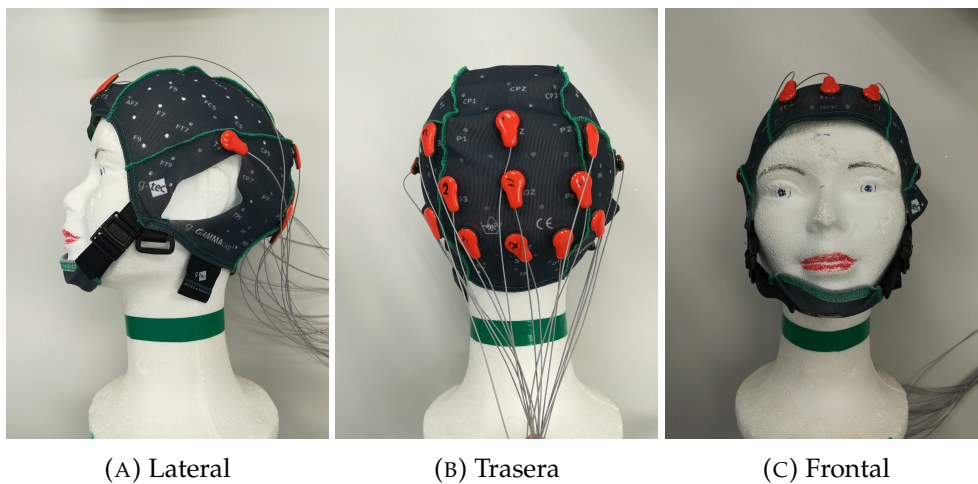


FIGURA 4.7: Imágenes de la malla de electrodos

#### 4.2.4.3 Dispositivos electrónicos

Como los electrodos elegidos son secos y activos necesitan de un preamplificador que proteja la débil señal a interferencias externas. Este dispositivo se llama g.SAHARAbbox

Y finalmente necesitamos el dispositivo más importante y más caro de este sistema, y es el que interconecta el Arduino con el ordenador y una toma de tierra para derivar la electricidad estática. Para llevar a cabo su función en el sistema se ha mantenido la configuración inicial del proyecto de (Rodríguez Muñoz et al., 2018) y consta de:

- **Frecuencia de Muestreo de 512Hz:** Más alta implicaría mas datos a procesar afectando a la carga computacional del algoritmo aquí presentado. Una frecuencia de muestreo baja, implica que se puedan perder armónicos con respecto a la frecuencia de estimulación. Por lo que el valor actual es el adecuado.





FIGURA 4.8: Preamplificador de electrodos

- **Filtrado de frecuencia determinada, 50Hz:** Es el filtrado de la frecuencia relacionada con el suministro eléctrico en España, y que es de 50Hz.
- **Filtrado bandpass 5Hz-60Hz:** Las frecuencias por debajo de 5Hz no están relacionadas con las señales SSVEPs y por encima de los 60Hz es muy difícil que su relación señal ruido sea útil, ya que a mayor frecuencia hay más ruido en la detección de SSVEPs.

Así mismo la alimentación eléctrica de todos los aparatos del BCI, incluido el ordenador y pantalla, están conectados a un sistema de protección eléctrica activa, que estabiliza la corriente y unifica la tierra común.



FIGURA 4.9: Amplificador de señal, USBamp.

### 4.3 Problemas solventados

Debido a que este proyecto se ha basado en el desarrollo de una plataforma Software, ha implicado el uso de tecnologías actuales y por tanto ha requerido de un rápido aprendizaje en un corto periodo de tiempo. Durante la fase de diseño y desarrollo



han surgido algunos problemas cuya solución puede ser útil para los colaboradores y que se expone en los siguientes puntos:

#### 4.3.1 Globales

- **Dimensionar el proyecto.** A todos nos ha pasado que el *brainstorming* es mucho más fácil y rápido que la implementación de las ideas. Dimensionar el proyecto significa estimar cuánto tiempo nos va a costar llevar a cabo todos los objetivos del proyecto en un tiempo determinado. Hay muchos aspectos en los que trabajar con los BCIs, pero hay que tomar la firme decisión de acotar la funcionalidad.
- **Diferencias entre Sistemas operativos.** El ordenador del laboratorio del Grupo de Neurocomputación Biológica (GNB) que usa el BCI tiene un Sistema Operativo Ubuntu 16.04. Por lo tanto se recomienda desarrollar en el mismo sistema para evitar sorpresas, aunque Python es multiplataforma, la gestión de permisos de las carpetas, conexión por el puerto USB y otros aspectos difieren de un sistema a otro. Incluso la instalación de librerías en cada sistema es diferente y el repositorio de dónde se descargan también.
- **Lenguajes de programación.** Los lenguajes que se han usado en este proyecto han sido Python, C++ y QT. Cada lenguaje tiene su propia sintaxis y forma de hacer las cosas. Además para validar la plataforma se han desarrollado scripts en Python y MATLAB para comprobar que los resultados son iguales al obtenido por (Rodríguez Muñoz et al., 2018) con la base de datos actual.
- **Desarrollo de la plataforma.** La plataforma es un sistema que incluye: la estimulación de LEDs, el análisis de señales y una interfaz gráfica que permite al usuario con conocimientos básicos en la materia controlar el flujo de trabajo complejo. Integrar todas las partes requiere de una visión global refinada para que pueda ser entendida de forma intuitiva.

#### 4.3.2 En implementación de algoritmos:

- **Versión de librerías.** La gestión de las librerías requiere especial cuidado, una actualización inadecuada puede causar alguna incompatibilidad entre ellas. Como ha pasado con la librería *Scipy* que tenía que ser en una determinada versión, y no la última, para que funcionara el coeficiente de Pearson.
- **CCA + coeficiente de correlación.** Una vez hemos conseguido la librería para el Análisis de Correlación Canónica necesitábamos algo más, solo con eso no iba a funcionar, se necesitaba el coeficiente de Pearson o alguno similar que busque la correlación de la salida de la función de CCA y que se ha conseguido con la librería *Sklearn*. En MATLAB la propia función *canoncorr* ya disponía de esa funcionalidad. Pero en este proyecto se ha decidido no usar MATLAB por su alto coste de licencia y poca flexibilidad con librerías modernas.
- **Interacción con el SDK de g-Tec.** El sistema SDK que ofrece g-Tec tenía asociado un script en MATLAB para la lectura de los ficheros binarios. Esta lectura se hacía de una forma poco común. Se leía un fichero binario escrito en little endian y con datos codificados de forma hexadecimal y en grupos de cuatro bytes. La traducción a script Python supuso una investigación a bajo nivel y una vez conseguida se integraba en la plataforma.

### 4.3.3 En Interfaz gráfica

- **Refresco en la ventana.** El refresco de la ventana no es automático. Para solventar este problema se ha refactorizado los comandos de forzado de actualización del entorno gráfico a una función para después ser llamada en los distintos pasos del flujo de trabajo.
- **Separar interfaz de la lógica.** Es difícil integrar la estimulación, análisis de señales y control de flujo en una interfaz sencilla. Por eso se tomó la decisión de separar en un fichero aparte toda aquella funcionalidad que se refiere al análisis de señales (*main.py*), en otro la interfaz (*screen\_main.py*) y en otro la funcionalidad de flujo de trabajo (*launcher.py*)
- **Todo es dinámico.** Uno de los aspectos más laboriosos en este trabajo es la flexibilidad que se le deja al usuario para realizar la configuración de parámetros en la interfaz y lo que ello implica en la plataforma. Se deja libertad tanto con el número de trial, mejores frecuencias, fases y electrodos, como otros parámetros que el usuario considere oportuno a criterio personal. Esto permite al usuario hacer distintas pruebas con distintos modelos tanto de localización de electrodos como sistema de validación final.

### 4.3.4 En validación

- **MATLAB vs Python** Es igual de importante realizar un algoritmo como la comprobación de que el resultado calculado es el correcto. En el trabajo de (Rodríguez Muñoz et al., 2018) se proponía un procesado desde MATLAB pero como ya se ha explicado, en este trabajo se ha usado Python como lenguaje principal. Gracias a que se disponía de la base de datos en el Grupo de Neurocomputación Biológica (GNB) se ha podido realizar las validaciones en la detección de señal SSVEPs mediante la comparación de resultados. Esto permite dos cosas, por un lado dar la seguridad que el algoritmo está bien desarrollado y por otra tener que aprender y modificar los script en MATLAB para adecuarlo a las pruebas de validación de resultados.

## Capítulo 5

# Resultados

### 5.1 Resultados de validación de la plataforma

Como se ha comentado en el capítulo de introducción, el objetivo de este TFM es el de realizar una plataforma para la implementación de experimentos de BCI basados en SSVEP que se adapte al usuario unificando distintas funcionalidades de selección de frecuencias, fases y electrodos mediante una interfaz gráfica que facilite su uso y distribución. Su desarrollo ha sido casi enteramente implementado en Python, exceptuando los comandos BASH y el SDK de g-Tec. Como todo desarrollo, necesita ser probado por partes y de forma global. Por ello se van a describir las distintas pruebas que validarán tanto su funcionalidad como la eficacia de la integración de los distintos algoritmos que utiliza.

#### 5.1.1 Control de la estimulación LED

Como se observa en la Figura 3.2 y 3.3, la arquitectura de conexiones es simple. Todo comienza con la acción de un usuario en la interfaz gráfica y que desencadena una serie de comandos internos que finalmente acaba con la activación de los LEDs. Dicha estimulación se realiza a través de la plataforma, la cual controla el flujo del proceso. Para la validación de esta parte desarrollada se deben ejecutar una serie de pasos en forma secuencial y que se detallan en las siguientes subsecciones.

##### 5.1.1.1 Secuencia de comandos de estimulación

A continuación se van a explicar brevemente los pasos de las líneas de código que sirven para controlar el proceso de estimulación de LEDs.

1. **Modo configuración.** Se ejecuta un comando que envía el número 179 al Arduino y recibe el echo si lo recibe. De esta forma sabemos que el Arduino ha recibido el comando que lo prepara para ser configurado.  
`> sudo echo "179" > PuertoArduino`
2. **Envío de parámetros.** Se configura la frecuencia y fase de cada LED y se envía al Arduino. Las frecuencias que están en 0.0 Hz significará que el LED permanecerá apagado, en caso contrario, hará el parpadeo cuando se le diga.  
`> sudo echo frec1 fase1 frec2 fase2 frec3 fase3 frec4 fase4 > PuertoArduino`
3. **Sincronización y grabación.** Se ordena al g.USBamp activar la grabación del registro de EEG. Éste a su vez espera 1 segundo y activará una conexión con el Arduino que comenzará a encender los LEDs con los parámetros anteriormente configurados.  
`> sudo script_g_tec`

4. **Apagado de LEDs.** Cuando el g.USBamp ha acabado de grabar, la plataforma manda un comando al Arduino para apagar todos los LEDs, puesto que la prueba ha terminado.

> *sudo echo "999" > PuertoArduino*

```
#Mode set on
execute('sudo echo "179" > '+ str(ARDUINO_PORT)+'')

#Configure Arduino
f = str(list_freq_to_test[current_freq_pos])
p = str(list_freq_phase[current_freq_pos])
execute('sudo echo "'+f+' '+p+' 0.0 1.5 0.0 1.5 0.0 1.5" > '+ str(ARDUINO_PORT)+'')

#gUSBamp trigger the led and record
err,out = execute('sudo echo $$&')
execute('sudo ./gUSBamp_files/gUSBampAPI_Demo '+str(out),showresult=False)

#Turn off leds
execute('sudo echo "999" > /dev/ttyACM0')
```

FIGURA 5.1: Secuencias de comandos de estimulación

```
def execute(str_command, showresult=False):
    if(showresult):
        print('EXECUTING:',str_command)
        result = subprocess.run(str_command, stdout=subprocess.PIPE, stderr=subprocess.PIPE, shell=True)
        sleep(0.25)
        err = result.stderr.decode("utf-8")
        out = result.stdout.decode("utf-8")
        if(showresult):
            print('Return code:',result.returncode)
            print('error:',err)
            print('output:',out)
    return err, out
```

FIGURA 5.2: Función desarrollada en Python que ejecuta comandos de Linux

Se ha refactorizado la función de código que hace llamadas por línea de comandos desde Python y que está integrado en la plataforma. De esta forma se consigue no solo claridad en el código sino que a la hora de hacer cambios solo se hace en un sitio.

### 5.1.2 Validación de la estimulación y el registro simultáneo de EEG

Vimos en el apartado anterior la secuencia de comandos necesaria para completar el proceso de estimulación, el cual inicia el usuario desde la interfaz gráfica de la plataforma. Procederemos en este apartado a detallar cómo se va tratando la información en la señal a través de todos los pasos. Antes de continuar hay que comentar que este sistema BCI usa 16 electrodos: PO7, PO3, POz, PO4, PO8, O1, Oz, O2, P3, PZ, P4, FP1, FP2, T7, T8 y FPZ. Cada uno de los cuales registra 6 segundos, 1 segundo sin estímulo y posteriormente 5 segundos con estímulo. Esta selección de electrodos es común a muchos experimentos de BCIs basados en la detección de SSVEPs, pero que puede ser fácilmente configurable.

### 5.1.2.1 Señal de estímulo

La señal de estimulación comienza en el Arduino, que es el responsable encender y apagar físicamente los LEDs. Genera una señal sinusoidal a una determinada frecuencia y fase. En la figura 5.3 se representa un ejemplo de validación en el que se ve cómo a partir del segundo 1 empieza a generar la señal estímulo y por lo tanto a parpadear el LEDs. Se ilustran dos señales generadas a diferentes frecuencias y fases.

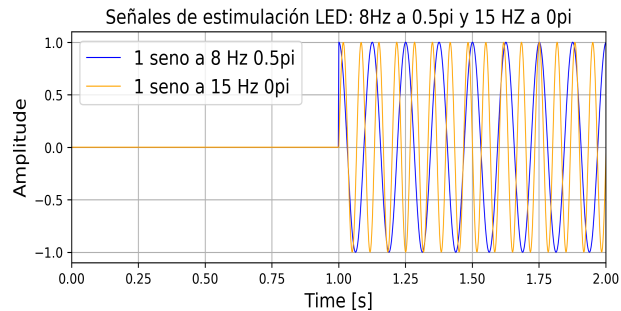


FIGURA 5.3: Señales de estimulación del Arduino

### 5.1.2.2 Validación en el registro del estímulo en la señal de EEG

Un vez ha empezado la estimulación LED, el usuario pone la atención visual al LED que ofrece el estímulo visual. Entonces el cerebro del usuario produce una respuesta fisiológica al estímulo generando una señal registrable con un EEG.

En la figura 5.4 representa un registro EEG en el electrodo OZ. Como se puede ver hay una línea roja vertical que indica el momento exacto en el que la estimulación LED empieza a funcionar, puede apreciarse incluso una disminución en amplitud de la señal registrada.

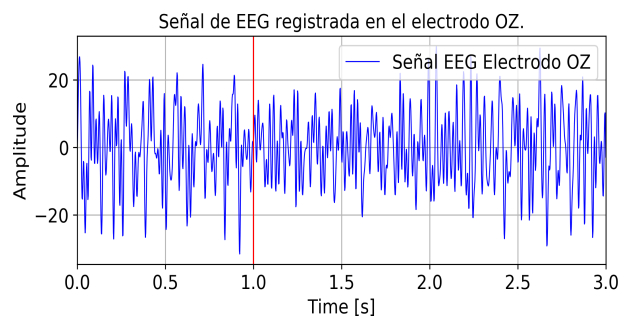


FIGURA 5.4: Ejemplo de validación del registro de EEG en el electrodo OZ

### 5.1.2.3 Análisis de la respuesta al estímulo en la señal de EEG

Para observar visualmente el efecto de la estimulación en el registro de EEG, se puede restar la señal adquirida por dos de los electrodos cercanos para eliminar el ruido común como ilustra la figura 5.5.

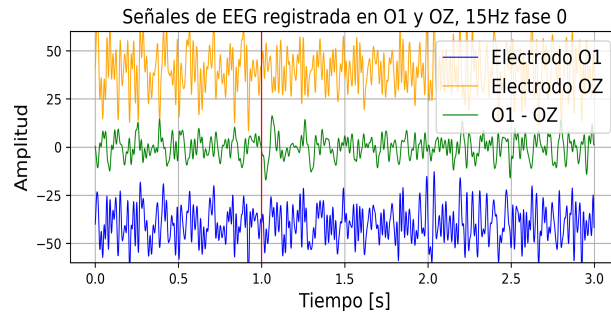


FIGURA 5.5: Diferencia entre señales procedente de electrodos cercanos.

Ahora para validar el registro del efecto del estímulo, vamos a sincronizar en una gráfica la señal estímulo que genera el Arduino y la diferencia de señales entre los electrodos obtenida anteriormente. El resultado se muestra en la figura 5.6.

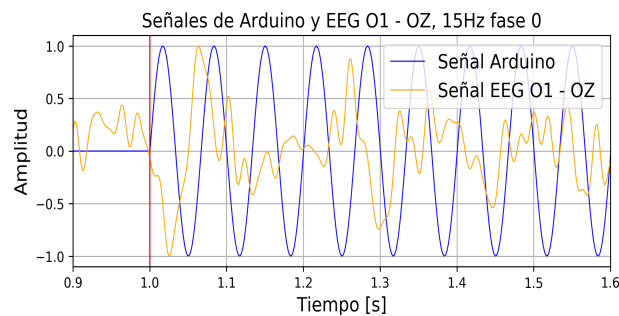


FIGURA 5.6: Sincronización señal estímulo del Arduino y la registrada en el EEG.

Vemos que hay cierta relación entre la señal que genera el Arduino con la que se registra en el EEG. Aun así, no podemos determinar claramente si es la señal que buscamos, por lo que necesitamos un método de análisis de señal más preciso.

#### 5.1.2.4 Análisis del registro EEG en frecuencias

Antes de comenzar con el análisis de señales y detectar la posible señal SSVEPs, es necesario transformar la señal del dominio del tiempo al dominio de frecuencias mediante la transformada rápida de Fourier (FFT). En este nuevo espacio, a veces se puede ver de manera visual cuál es la frecuencia estímulo y otras no.

Como vemos en la figura 5.7, si optamos por la frecuencia de estimulación de 15 Hz somos capaces a simple vista de detectar cuál ha sido el estímulo. En cambio, si elegimos la frecuencia de 13 Hz vemos que hay una serie de picos de amplitud similar a la del estímulo como se puede apreciar en la figura 5.8

Para solventar esto necesitamos de un análisis de señales más avanzado y para ello usaremos el Análisis de Correlación Canónica (CCA) el cual utiliza la señal de referencia, es decir, la del estímulo para generar un conjunto de senos y cosenos a comparar con la señal registrada por los electrodos. Veremos los resultados en el siguiente apartado.

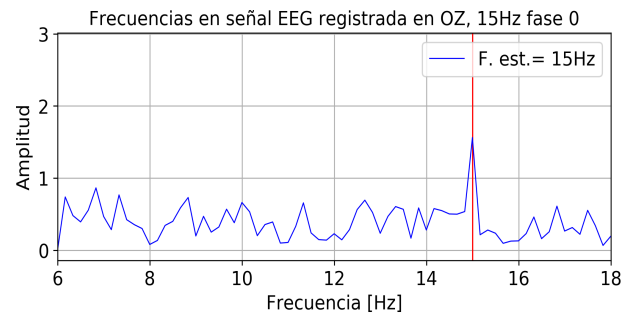


FIGURA 5.7: Demostración de detección visual de la frecuencia de estímulo

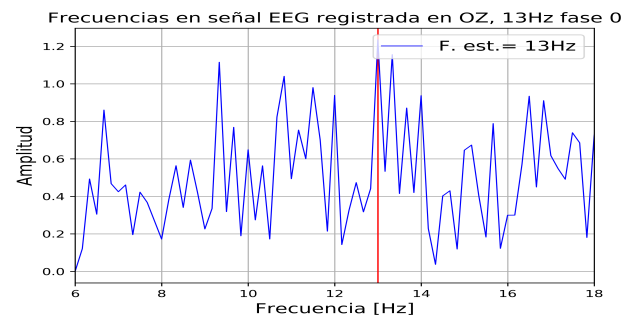


FIGURA 5.8: Ejemplo de la problemática que se plantea en algunos casos para decodificar el estímulo en la señal de EEG cuando hay varios picos de amplitud parecida en el análisis de la transformada de Fourier

### 5.1.3 Validación de detección señales SSVEPs con un estímulo

En las imágenes de la figura 5.9 se puede observar cuál es el resultado de la estimulación a una determinada frecuencia y su correspondiente detección a través del Análisis de la Correlación Canónica (CCA). Como la interfaz permite seleccionar un rango de frecuencias, en este ejemplo las frecuencias van de 8Hz a 15Hz. Como se puede comprobar la detección se cumple en todos los casos.

Se puede ver además en la figura 5.9 que en las subfiguras con la frecuencia 10 y 13 Hz han aparecido otros picos, pero el Análisis de Correlación Canónica (CCA) hace una correcta detección, como se observa en todas las imágenes mostradas.

Estas gráficas han sido generadas automáticamente desde la propia plataforma. Para ello basta con llamar a la función `plot_fft_signals` e indicar el fichero de grabación, frecuencia de estímulo y número de trial. La función procesará el fichero, hará una transformada rápida de Fourier (FFT) de la señal registrada en el electrodo OZ, realizará un Análisis de la Correlación Canónica (CCA) y mostrará la gráfica con una franja y una línea vertical mostrando la frecuencia detectada.

### 5.1.4 Validación de detección señales SSVEPs con cuatro estímulos

Como sabemos la estimulación en el apartado anterior era de un solo estímulo a la vez, es decir, la plataforma estimulaba a una frecuencia determinada y procesaba la señal registrada en el EEG con el Análisis de Correlación Canónica. Pero en la vida real con la aplicación de los BCIs se utilizan más de un estímulo para realizar distintas acciones, por lo que es necesario validar que la plataforma es

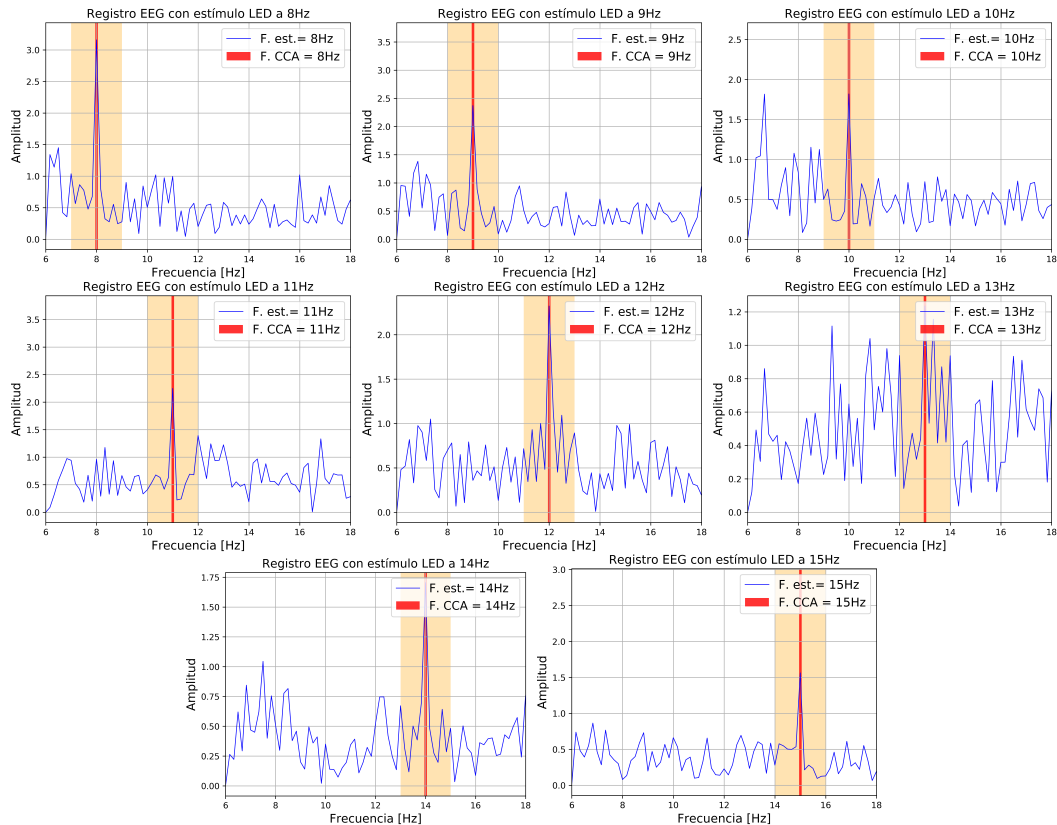


FIGURA 5.9: Imagen de detección automática de señales SSVEPs desde la plataforma Software, un estímulo a la vez.

capaz de trabajar con cuatro estímulos a la vez y comprobar que el análisis detecta correctamente la frecuencia estimulada.

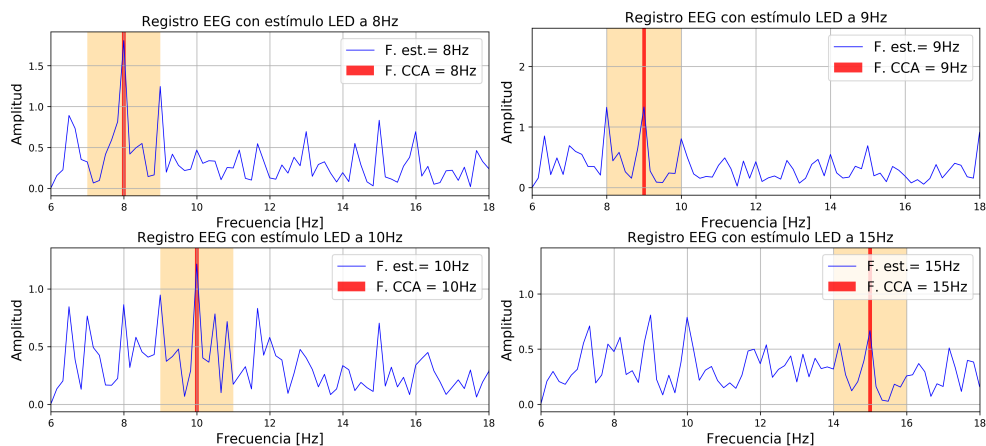


FIGURA 5.10: Imagen de detección automática de señales SSVEPs desde la plataforma Software, cuatro estímulos a la vez.

Comparando la figura 5.9 y la 5.10, podemos comprobar que una estimulación conjunta tiene una respuesta fisiológica distinta por parte del usuario a cuando se estimular por separado. Si nos fijamos en la frecuencia 15Hz de la figura 5.9 se puede ver claramente dónde está el estímulo. En cambio en la figura 5.10 y con la misma frecuencia vemos que aparecen una serie de frecuencias con mayor amplitud



incluso que la del estímulo. Vemos además que la plataforma detecta perfectamente la señal del estímulo cuya amplitud no es la mayor.

### 5.1.5 Validación de las distintas etapas de un experimento

En esta sección se van a validar las distintas etapas de las que se compone el experimento, para ello se utiliza la interfaz gráfica de la plataforma.

#### 5.1.5.1 Datos de sesión y usuario

En primer lugar se deberá rellenar unos campos que se refieren al usuario que llevará el casco de electrodos. Estos datos son totalmente anónimos por protección de datos. Internamente se puede llevar una lista de qué nombre de usuario corresponde a qué persona física con su correspondiente encriptación.

Se rellenan los campos de nombre de usuario, género, si lleva gafas, la edad, y algún comentario. Además hay que escribir el nombre de la sesión y seleccionar la fecha. La concatenación de estos dos últimos campos será el código de sesión que será único y se usará para la carpeta que contiene todos los ficheros generados en la sesión.

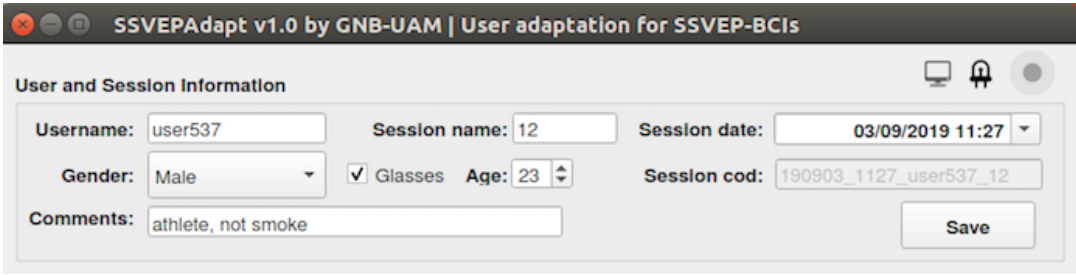


FIGURA 5.11: Interfaz: sección de usuario y sesión

Todos estos campos son objeto de ser almacenados en una base de datos. Y para ello la plataforma permite una exportación a fichero JSON, el cual permite guardar la información en una estructura estandarizada. Se verá en detalle en la subsección de Validación de la extracción.

#### 5.1.5.2 Selección de Modelo de localización de electrodos

El modelo de localización y conjunto de electrodos es lo primero que hay que configurar antes de empezar con la estimulación. El modelo internacional 10-20 en la figura 4.6 es ampliamente usado, al igual que la configuración que se ha presentado en la plataforma para el conjunto de electrodos y que se ajusta en su mayoría a la zona occipital. Se ha determinado un conjunto de 16 electrodos que se han utilizado en todos los experimentos de validación: PO7, PO3, POz, PO4, PO8, O1, Oz, O2, P3, PZ, P4, FP1, FP2, T7, T8 y FPZ. En caso de que se quiera utilizar una configuración distinta se podría indicar en la zona de comentarios de la sección de Usuario y Sesión.

Para hacer más flexible la plataforma se puede incluir un pequeño fichero de configuración, por ejemplo *config.ini*, en el cual se añadan tantas configuraciones como sean necesarias. Luego en la interfaz de la plataforma sería seleccionable en

la lista. Esto permitiría al usuario de la interfaz ampliar el tipo de estudios en la detección de SSVEPs con otra configuración de electrodos.

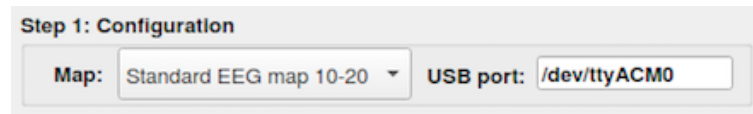


FIGURA 5.12: Interfaz: sección de modelo

### 5.1.5.3 Validación en la selección de frecuencias de estímulo y trials

La plataforma permite seleccionar un rango de frecuencias en el cual se ha optado por un incremento determinado de 1 Hz. No obstante, se podrían explorar todo un abanico de incrementos de la frecuencia de un extremo al otro del rango, pero por conveniencia se ha tomado la decisión por una en particular. El número de trials o de repeticiones de barrido de frecuencias también se puede seleccionar. Además dispone de un botón que lanzará todo el proceso de estimulación. Haciendo click en el botón, hará tantas estimulaciones LEDs, análisis de señales y detección online de SSVEPs como frecuencias existan en el rango de barrido. En resumen, si tenemos 2 trials con una lista de 8 frecuencias(8, 9, 10, 11, 12, 13, 14 y 15), solo harán falta dos clicks y hará todas las estimulaciones, esto es, 8 frecuencias x 2 trials.



FIGURA 5.13: Interfaz: sección de frecuencia y electrodos

Una vez le hayamos dado 2 veces al click, porque hemos configurado 2 trials, tendremos que dar un click extra más que desencadenará la detección de SSVEPs offline mediante el CCA y usará todas las grabaciones generadas. El resultado será una lista de los mejores electrodos, ordenados de mayor a menor importancia; y una lista de las mejores frecuencias, ordenadas de mayor a menor ratio.

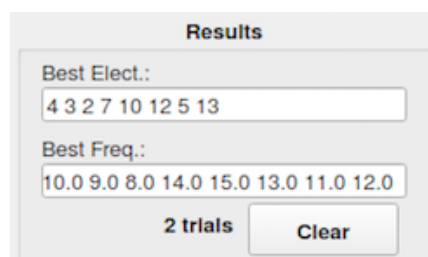


FIGURA 5.14: Interfaz: resultado de la frecuencias y electrodos

Como se puede observar en la figura 5.14 solo contiene la primera mitad de los electrodos de la tabla 5.2, los cuales están ordenados en función de su importancia en el ratio RCC calculado. Una posición alta en la tabla indica un bajo ratio RCC

Frecuencia Hz	Num. aciertos	RCC
10.0	2	4.3719
9.0	2	3.7885
8.0	2	3.2567
14.0	2	2.1373
15.0	2	1.9853
13.0	2	1.9518
11.0	2	1.9332
12.0	1	2.0863

TABLA 5.1: Tabla resultado de las mejores frecuencias

Pos. electrodo	Nom. electrodo	RCC
4	PO8	4.2767
3	PO4	4.3209
2	POZ	4.4028
7	O2	4.4077
10	P4	4.438
12	FP2	4.4434
5	O1	4.4759
13	T7	4.4831
6	OZ	4.4876
14	T8	4.4886
11	FP1	4.5152
15	FPZ	4.5251
9	PZ	4.5288
1	P03	4.5652
8	P3	4.5943
0	P07	4.655

TABLA 5.2: Tabla resultado de los mejores electrodos

al quitar ese electrodo, lo que significa que con su inclusión se obtiene una mejor detección de señales SSVEPs.

Para la realización de los mejores electrodos optaremos pues con la primera mitad de la lista y en caso de que el usuario de la interfaz no obtenga buenos resultados podrá optar por otra configuración.

#### 5.1.5.4 Validación en la selección de Frecuencia, rango de fases y trials

Ya tenemos dos parámetros importantes: los electrodos y las frecuencias, falta determinar la fase. Para ello cogemos la mejor frecuencia y aplicamos un criterio personal sin estimulación, basándonos en que las fases deben permitir una mejor separación entre estímulos cuando se estimulan todos los LEDs a la vez. El procedimiento es el siguiente: escogemos la mejor frecuencia y tres más, en total cuatro porque tenemos cuatro LEDs. Se aplica una regla a la mejor frecuencia y obtenemos una fase en concreto, posteriormente aumentamos en  $0.5\pi$  a cada frecuencia siguiente de tal forma que la lista de fases oscila entre  $1.5\pi$  y  $0.5\pi$ .

Nuevamente tenemos el botón para hacer el cálculo de fases, por lo que deberemos pulsarlo para realizar este proceso.

Como se comentaba antes en la forma de asignar las fases, vemos que después de llegar al  $1.5\pi$  vuelve al  $0\pi$ . Y el resultado es el siguiente.

Step 3: Calibration - Best Phases

Freq.: 10.00 From: 0.00 To: 1.50 Trials: 1 >

FIGURA 5.15: Interfaz: sección de fase

Best Phase:

1.0 1.5 0.0 0.5

FIGURA 5.16: Interfaz: resultado de la fase

#### 5.1.5.5 Validación con los mejores parámetros

Para realizar este paso, el procedimiento es el siguiente: se dispone de una lista que en esta versión solo ofrece un tipo de validación, y es la estimulación de 4 LEDs con la configuración que se disponga. Hay unos radiobuttons que si están seleccionados en *Electrodos*, *Frecuencias* o *Fases* obtendrá automáticamente lo calculado anteriormente. En caso contrario, si se elige *Otro*, habrá que especificar cuáles son. Esto último es muy importante, ya que la plataforma está pensada para que el usuario de la interfaz pueda hacer un estudio sobre la adaptabilidad de parámetros a usuarios del BCI en la detección de señales SSVEPs. En resumen, si los mejores parámetros no tienen un buen resultado, el usuario de la interfaz podrá cambiarlo. Figura 5.17

Step 4: Validation

Application: Online Test Applicaton Trials: 2 >

Using best:

- ☒ Electrodes ☐ Other 4 3 2 7 10 12 5 13
- ☒ Frecuencias ☐ Other 10.0 9.0 8.0 14.0
- ☒ Phases ☐ Other 1.0 1.5 0.0 0.5

FIGURA 5.17: Interfaz: sección de validación

Una vez se hayan terminado todos los trials y dado al click extra para el cálculo offline, se realizará nuevamente un análisis completo dando como resultado una sección de las mejores frecuencias. Figura 5.18

Comparando la figura 5.14 con la 5.18 vemos que la primera tiene un orden diferente a la segunda. Ambas contienen las mejores frecuencias, pero cuando se han estimulado a la vez el resultado del análisis ha determinado un nuevo orden. Esto que puede parecer de poca relevancia pero realmente la tiene pues podemos deducir que la respuesta fisiológica del usuario ha cambiado, aunque no es objetivo del estudio en este trabajo.

**Results**

Detected Frequencies:  
9.0 8.0 10.0 14.0

2 trials Clear

Final Score: 100.0 %

Export to JSON

FIGURA 5.18: Interfaz: resultado de la validación

Frecuencia Hz	Num. aciertos	RCC
9.0	2	3.5093
8.0	2	2.3447
10.0	2	2.328
14.0	2	1.9437

TABLA 5.3: Tabla resultado de las mejores frecuencias estimuladas en conjunto

Existe un sistema de puntuación en la plataforma que determinará cómo de buena ha sido la detección de señales SSVEPs en el experimento. La puntuación mide cuántos aciertos han habido con respecto al máximo posible, es decir, *número de trials \* número de LEDs*. En el ejemplo de arriba, podemos ver que la puntuación es del 100%. En caso de que el porcentaje no sea el deseado, el usuario de la interfaz podrá reconfigurar los parámetros y repetir el experimento.

Vamos a realizar otra prueba seleccionando una configuración distinta a la mejor calculada anteriormente. Para ello vamos a seleccionar la segunda mitad de la tabla de electrodos y frecuencias, dejando las fases a  $0 * \pi$  de todas las frecuencias. Ejecutamos solo la etapa de validación conjunta y realizamos varias estimulaciones.

**Step 4: Validation**

Application: Online Test Applicator Trials: 2 >

Using best: ☐ Electrodes ☒ Other 6 14 11 15 9 1 8 0

☐ Frecuencias ☒ Other 15.0 13.0 11.0 12.0

☐ Phases ☒ Other 0.0 0.0 0.0 0.0

FIGURA 5.19: Interfaz: sección de validación alternativa

Como vemos en la figura 5.20 el porcentaje de acierto ha bajado debido a que en la cantidad de aciertos en la frecuencia 15 Hz ha bajado.

Es importante recalcar que a menos trials o barrido de estímulos la granularidad en el detalle de acierto es menor, por lo tanto conviene buscar un número adecuado a las necesidades del estudio y la disponibilidad de los usuarios paciente.

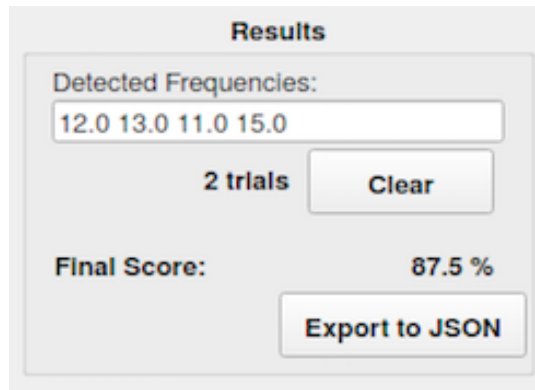


FIGURA 5.20: Interfaz: resultado de la validación alternativa

Frecuencia Hz	Num. aciertos	RCC
12.0	2	2.9267
13.0	2	2.6815
11.0	2	2.2154
15.0	1	2.0268

TABLA 5.4: Tabla resultado de frecuencias en una configuración alternativa.

Esta prueba se ha realizado para validar que la plataforma realiza todo el proceso correctamente en todas sus etapas: estimulación de LEDs, registro de EEG, análisis de señales mediante CCA y permitiendo libertad en la configuración para la realización de las pruebas. Además, con los resultados obtenidos se ha validado la efectividad en el proceso y el sistema de puntuación para determinar cuanto de cerca han estado los resultados de las pruebas con el objetivo deseado. Y como se ha explicado en sección anteriores, todo ello es posible a través de la interfaz gráfica que incluye la plataforma, que facilita en gran medida el uso de este BCI por personas ajenas al Software o la Electrónica.

### 5.1.6 Validación en la extracción de datos del experimento

Hemos visto que en las distintas etapas del experimento se ha ido generando una valiosa información que recoge todos los parámetros importantes. Además hemos visto que en la figura 5.10 la plataforma proporciona una forma de exportar toda la información generada en formato JSON. El objetivo es que sea usado como forma de importación en una base de datos de terceros y todo ello de forma anónima ocultado la verdadera identidad del usuario.

Además se conservarán los archivos originales que la plataforma ha generado en cada estimulación y con el formato establecido en el apartado de Estructura de ficheros en el capítulo de Desarrollo de este trabajo. Esto permite un análisis offline en cualquier momento favoreciendo la validación de nuevas técnicas de análisis de señales.

### 5.1.7 Modalidad en las pruebas

En el apartado anterior hemos visto en detalle toda la funcionalidad que la interfaz de la plataforma permite hacer. Pero también es posible realizar combinaciones de

```
"user": [  
  {  
    "username": "user537"  
  },  
  {  
    "gender": "Male"  
  },  
  {  
    "age": 23  
  },  
  {  
    "glasses": true  
  },  
  {  
    "sessio_name": "12"  
  },  
  {  
    "sesion_cod": "190903_1127_user537_12"  
  },  
  {  
    "comments": "athlete, not smoke"  
  },  
  {  
    "best_elec": "4 3 2 7 10 12 5 13 "  
  },  
  {  
    "best_freq": "10.0 9.0 8.0 14.0 15.0 13.0 11.0 12.0 "  
  },  
  {  
    "times_elec_and_freq": 2  
  },  
  {  
    "best pha": "1.0 1.5 0.0 0.5 "  
  },  
  {  
    "times pha": 0  
  },  
  {  
    "best_online_freq": "9.0 8.0 10.0 14.0 "  
  },  
  {  
    "times_online": 2  
  },  
  {  
    "online_score": "100.0 %"  
  }  
]
```

FIGURA 5.21: Interfaz: exportación en JSON de los parámetros del experimento

pasos de forma no secuencial para establecer comparaciones de eficacia. Las posibles modalidades son las siguientes:

- **Validación sin adaptación al usuario:** Básicamente es realizar la validación final sin los pasos previos de obtención de los mejores parámetros. Esto permite hacer una prueba rápida para determinar, por ejemplo, si existe el BCI Illiteracy en el usuario, o si el sistema está bien montado y listo para funcionar.

- **Validación con adaptación al usuario:** Como se explicó en el apartado anterior, consiste en seguir los pasos de la interfaz de forma secuencial. Con ello se comprobará la compatibilidad de la adaptación al usuario y con las frecuencias en conjunto.
- **Validación con adaptación comparable:** Es la modalidad recomendable y podría determinar si se consigue una mejora o no con la adaptabilidad al usuario en el BCI. En caso de que los resultados no sean favorables, el usuario de la interfaz puede repetir las pruebas con distinta configuración.

Es importante comentar que las pruebas causan cierto malestar y en distinta medida a los usuarios del BCI, debido a la naturaleza de la propia estimulación. Por lo que las pruebas no pueden ser infinitas y hay que acotarlas de algún modo. Además el cansancio o el aprendizaje del propio cerebro son factores que van en contra de las pruebas. Por lo tanto el tiempo de las pruebas es determinante.



## Capítulo 6

# Conclusiones y Futuro Trabajo

### 6.1 Conclusiones

La principal motivación de este trabajo ha sido la de integrar en una plataforma Open Source la estimulación de LEDs con señal sinusoidal y detección de señales SSVEPs con Análisis de Correlación Canónica (CCA) con la capacidad de adaptarse al usuario mediante la selección de fases, frecuencias y electrodos.

La adaptación al usuario mediante la selección de frecuencias, fases y electrodos conduce a una mejora de la eficacia de un interfaz cerebro-máquina y al desarrollo de interfaces asistidos (Fernandez-Vargas et al., 2013b). Anteriormente esta mejora se ha demostrado por separado (Fernandez-Vargas et al., 2013a; Rodríguez Muñoz et al., 2018; Wu and Su, 2014; Bin et al., 2009), pero no conjuntamente hasta el momento. La plataforma diseñada y validada en este trabajo permitirá recoger los datos para realizar dicha validación conjunta.

Para poder trabajar con esta plataforma, se presenta con una interfaz gráfica de usuario amigable y que un usuario con conocimientos limitados en la materia pueda hacer distintos experimentos. Se optó por lenguajes de programación sin licencia de pago, como es el C++, Python y QT para la interfaz gráfica. Además se han automatizado procesos internos, de tal forma que haciendo un click en la interfaz gráfica desencadena un barrido de estimulaciones, un análisis rápido online mediante el CCA, y finalmente un análisis offline de todas las grabaciones. Se ha diseñado y desarrollado una interfaz compacta y sencilla, reduciendo a pocas acciones por parte del usuario.

El objetivo más allá de los planteados y cumplidos en este trabajo es que se pueda estudiar en profundidad y de una forma fácil la adaptación al usuario de los mejores parámetros para controlar un BCI, ya sea una silla de ruedas, un exoesqueleto u otros sistemas. Como es de esperar, los principales usuarios de los BCIs son aquellos que tienen problemas neurodegenerativos y no pueden realizar actividades motoras. Los usuarios de la interfaz gráfica pueden usar esta plataforma para descubrir esa adaptabilidad de los parámetros del BCI al usuario paciente.

Hemos visto en la revisión del estado del arte que existen distintos estudios con metodologías de diversa complejidad pero que ninguno consigue el 100% de precisión y fiabilidad. El BCI adaptado al usuario sigue siendo en gran medida objeto de investigación activa, con este trabajo se espera que instituciones de diversa naturaleza puedan hacer estudios y combinar conocimiento, incluyendo el uso de la estimulación SSVEP en nuevas aplicaciones clínicas como sugieren modelos recientes (Latorre et al., 2019). Además se ha diseñado y desarrollado la plataforma para que futuros colaboradores pueden contribuir con nuevas técnicas y modificaciones. Esta plataforma es por tanto un punto común desde el que comenzar una nueva línea de investigación conjunta.

## 6.2 Futuro Trabajo

Basándome en la revisión del arte y la dimensión que se le ha dado a este trabajo, el futuro trabajo se puede categorizar en dos: interfaz gráfica de usuario y funcionalidad.

### 6.2.1 Interfaz gráfica de usuario

- **Interfaz gráfica móvil:**

Una mejora interesante es la adición de una segunda interfaz gráfica para usos móviles. Con ello no se necesita estar delante de un ordenador para controlar la plataforma. Un pequeño ordenador hace de servidor de plataforma conectado al BCI pero sin pantalla, y un móvil podría interactuar con una interfaz Web conectado de forma inalámbrica y usando servicios web.

### 6.2.2 Funcionalidad

- **Modelos de posición de electrodos:**

Vimos que la selección de modelos de localización de electrodos en esta versión de la plataforma era estática, pero se pueden incorporar nuevos modelos con distintas configuraciones de emplazamiento de los electrodos.

- **Detección SSVEPs y P300:**

Sabemos que las señales P300 suelen tardar un poco más en aparecer en respuesta al estímulo y por tanto tienen menor tasa de transferencia de información por unidad de tiempo, pero también es cierto que la precisión puede verse afectada de forma positiva con su combinación en la detección de señales de SSVEP en lo que se conoce como BCI híbridos (Chang et al., 2016). La tecnología integrada en este proyecto permite la ampliación a este tipo de interfaces.

- **Redes Neuronales como apoyo estadístico para mejorar la detección de SSVEPs:** Sabemos que las redes neuronales artificiales tiene un coste computacional elevado y que los sistemas BCI online no se pueden permitir su uso en muchas aplicaciones. Pero la estadística puede guiar la adaptación al usuario calculándose cada cierto tiempo. Por ejemplo, para tener en cuenta cómo la respuesta fisiológica de un usuario puede cambiar a lo largo del día, un cálculo constante en background puede generar mapas de actuación o de parámetros en los que es más probable que suceda el evento. De esta forma, la variabilidad de la respuesta no afectaría al resultado del BCI.

- **El historial de acciones influye:**

Los pacientes demandan sistemas BCI rápidos y fiables. Como esto de momento está algo lejos de conseguirse, la plataforma podría tener en cuenta el historial y las acciones más comunes en las mediciones. Esto quiere decir que si normalmente un usuario del BCI hace una secuencia de acciones que siguen una lógica, la estadística basada en la información almacenada de acciones diría que algunas acciones o secuencia de acciones son más probables que otras. Esto podría mejorar la usabilidad y la eficacia del sistema BCI.

- **Sistema de votación:** Se podría añadir un sistema de votación que incluya distintas tecnologías para la detección de señales y la correspondiente

decodificación del comando. Este procesamiento se podría realiza de forma paralela y cuyo resultado final se vería acotado por la votación. De tal forma que si 2/3 técnicas han determinado una acción determinada, esta será la elegida por la plataforma como salida de comando.



# Bibliography

- Attia, M., Hettiarachchi, I., Mohamed, S., Hossny, M., and Nahavandi, S. (2018). A frequency domain classifier of steady-state visual evoked potentials using deep separable convolutional neural networks. In *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. 2134–2139. doi:10.1109/SMC.2018.00367
- Bin, G., Gao, X., Yan, Z., Hong, B., and Gao, S. (2009). An online multi-channel SSVEP-based brain–computer interface using a canonical correlation analysis method. *Journal of Neural Engineering* 6, 046002. doi:10.1088/1741-2560/6/4/046002
- Chabuda, A., Durka, P., and Żygierewicz, J. (2018). High frequency ssvep-bci with hardware stimuli control and phase-synchronized comb filter. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 26, 344–352. doi:10.1109/TNSRE.2017.2734164
- Chang, M. H., Lee, J. S., Heo, J., and Park, K. S. (2016). Eliciting dual-frequency ssvep using a hybrid ssvep-p300 bci. *Journal of neuroscience methods* 258, 104–113
- Chen, X., Wang, Y., Nakanishi, M., Gao, X., Jung, T.-P., and Gao, S. (2015). High-speed spelling with a noninvasive brain–computer interface. *Proceedings of the national academy of sciences* 112, E6058–E6067
- Chen, X., Wang, Y., Zhang, S., and Gao, X. (2018). Enhancing detection of ssveps with intermodulation frequencies using individual calibration data. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. 2531–2534. doi:10.1109/EMBC.2018.8512783
- Choi, I., Rhiu, I., Lee, Y., Yun, M. H., and Nam, C. S. (2017). A systematic review of hybrid brain-computer interfaces: Taxonomy and usability perspectives. *PloS one* 12, e0176674
- Fernandez-Vargas, J., Pfaff, H., Rodriguez, F., and Varona, P. (2013a). Assisted closed-loop optimization of ssvep-bci efficiency. *Frontiers in Neural Circuits* 7, 27. doi:10.3389/fncir.2013.00027
- Fernandez-Vargas, J., Pfaff, H. U., Rodríguez, F. B., and Varona, P. (2013b). Assisted closed-loops for brain-computer interfaces. *BMC Neuroscience* 14, P406. doi:10.1186/1471-2202-14-s1-p406
- Herrmann, C. S. (2001). Human EEG responses to 1-100 Hz flicker: resonance phenomena in visual cortex and their potential correlation to cognitive phenomena. *Experimental Brain Research* 137, 346–353. doi:10.1007/s002210100682
- Jasper, H. H. (1958). The ten-twenty electrode system of the international federation. *Electroencephalogr. Clin. Neurophysiol.* 10, 370–375

- Kalunga, E. K., Chevallier, S., and Barthélemy, Q. (2018). Transfer learning for ssvep-based bci using riemannian similarities between users. In *2018 26th European Signal Processing Conference (EUSIPCO)*. 1685–1689. doi:10.23919/EUSIPCO.2018.8553441
- Kandel, E. R., Schwartz, J. H., and Jessell, T. M. (2012). *Principles of Neural Science* (New York: McGraw-Hill Education), 5 edn.
- Latorre, R., Varona, P., and Rabinovich, M. I. (2019). Rhythmic control of oscillatory sequential dynamics in heteroclinic motifs. *Neurocomputing* 331, 108 – 120. doi:https://doi.org/10.1016/j.neucom.2018.11.056
- Lim, J.-H., Hwang, H.-J., Han, C.-H., Jung, K.-Y., and Im, C.-H. (2013). Classification of binary intentions for individuals with impaired oculomotor function: ‘eyes-closed’ ssvep-based brain–computer interface (bci). *Journal of neural engineering* 10, 026021
- Lin, K., Gao, S., and Gao, X. (2019). Boosting the information transfer rate of an ssvep-bci system using maximal-phase-locking value and minimal-distance spatial filter banks. *Tsinghua Science and Technology* 24, 262–270. doi:10.26599/TST.2018.9010010
- Martinez, P., Bakardjian, H., and Cichocki, A. (2007). Fully online multicommand brain-computer interface with visual neurofeedback using ssvep paradigm. *Computational intelligence and neuroscience* 2007
- Myers, B. A. (1998). A brief history of human-computer interaction technology. *Interactions* 5, 44–54. doi:10.1145/274430.274436
- Nicolas-Alonso, L. F. and Gomez-Gil, J. (2012). Brain Computer Interfaces, a Review. *Sensors* 12, 1211–1279. doi:10.3390/s120201211
- Rabinovich, M., Friston, K., and Varona, P. (2012). *Principles of brain dynamics: global state interactions* (Cambridge, MA: MIT Press)
- Ramadan, R. A., Refat, S., Elshahed, M. A., and Ali, R. A. (2015). Basics of brain computer interface. In *Brain-Computer Interfaces* (Springer). 31–50
- Ramadan, R. A. and Vasilakos, A. V. (2017). Brain computer interface: control signals review. *Neurocomputing* 223, 26 – 44. doi:https://doi.org/10.1016/j.neucom.2016.10.024
- Rico Maldonado, J. et al. (2018). *Adaptación personalizada en interfaces cerebro-máquina añadiendo información de frecuencia y fase*. B.S. thesis
- Rodríguez Muñoz, D. et al. (2018). *Optimización y personalización de interfaces cerebro-máquina basados en potenciales visuales evocados*. Master’s thesis
- Turkheimer, F. E., Leech, R., Expert, P., Lord, L. D., and Vernon, A. C. (2015). The brain’s code and its canonical computational motifs. From sensory cortex to the default mode network: A multi-scale model of brain function in health and disease. *Neuroscience and Biobehavioral Reviews* 55, 211–222. doi:10.1016/j.neubiorev.2015.04.014
- Vidaurre, C. and Blankertz, B. (2010). Towards a cure for BCI illiteracy. *Brain Topography* 23, 194–8. doi:10.1007/s10548-009-0121-6

- Waytowich, N., Lawhern, V. J., Garcia, J. O., Cummings, J., Faller, J., Sajda, P., et al. (2018). Compact convolutional neural networks for classification of asynchronous steady-state visual evoked potentials. *Journal of Neural Engineering* 15, 066031. doi:10.1088/1741-2552/aae5d8
- Weiskopf, N., Mathiak, K., Bock, S. W., Scharnowski, F., Veit, R., Grodd, W., et al. (2004). Principles of a brain-computer interface (bci) based on real-time functional magnetic resonance imaging (fmri). *IEEE transactions on biomedical engineering* 51, 966–970
- Wolpaw, J. R., Birbaumer, N., McFarland, D. J., Pfurtscheller, G., and Vaughan, T. M. (2002). Brain–computer interfaces for communication and control. *Clinical Neurophysiology* 113, 767 – 791. doi:https://doi.org/10.1016/S1388-2457(02)00057-3
- Wu, Z. and Su, S. (2014). A dynamic selection method for reference electrode in ssvep-based bci. *PLOS ONE* 9, 1–12. doi:10.1371/journal.pone.0104248
- Zhang, Z., Wang, C., Ang, K. K., Wai, A. A. P., and Nanyang, C. G. (2018). Spectrum and phase adaptive cca for ssvep-based brain computer interface. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. 311–314. doi:10.1109/EMBC.2018.8512267





## Appendix A

# Glosario

- **Arduino:** Pequeño sistema electrónico que hace de ordenador pero con limitadas características.
- **Closed-Loop System:** Modalidad en las Interfaces cerebro máquina (BCI) en la que se proporciona un feedback al usuario.
- **Código fuente:** Se refiere el formato original en el que es desarrollado un Software. Posteriormente se compila o interpreta y es ejecutado en un ordenador.
- **Electroencefalografía (EEG):** Técnica utilizada para la detección de señales cerebrales mediante el uso de unos electrodos localizados en una posición determinada.
- **Fase:** La fase de una señal es la diferencia que existe entre cuando se ha generado la señal a una de referencia en e tiempo, y se mide en múltiplos de  $\pi$ .
- **Feedback:** Se define como la retroalimentación que recibe el emisor del mensaje y que recibe de una respuesta del receptor.
- **Frecuencia:** La frecuencia de la señal es la cantidad de veces que oscila de su máxima a la mínima amplitud y se mide en Hz.
- **G-tec** Empresa que fabrica el equipamiento electrónico de para la detección de EEG.
- **Interfaz Cerebro máquina (BCI) o (BMI):** Sistema electrónico que junto con técnicas en detección de señales, tiene como objetivo la comunicación de personas con máquinas para controlar o ejecutar una acción.
- **Potenciales visualmente evocados de estado estacionario (SSVEP):** Señal que genera el cerebro como respuesta fisiológica a un estímulo visual externo que parpadea a una determinada frecuencia y fase.
- **Script:** pequeño desarrollo Software para una tarea concreta.
- **Software Open Source:** Artefacto Software que es de libre distribución y modificación por otros autores. Es ampliamente acogido por la comunidad de desarrolladores al ser gratuito y con soporte técnico en algunos casos.
- **SDK:** Conjunto de librerías que hacen de herramientas en un desarrollo determinado.



## Appendix B

# Apéndice

## B.1 Manual de instalación del entorno

Como ya se ha comentado en los capítulos de Diseño y Desarrollo la plataforma Software se conecta al dispositivo g.USBamp a través de una API desarrollada en C++ y proporcionada por la misma compañía. En las siguientes secciones se va a proceder a describir los pasos necesarios para poder ejecutar la plataforma de forma adecuada.

### B.1.1 API g.USBamp

Para conseguir la API tendremos que ir a la página web de la compañía g-Tec y descargarnos el Software relacionado con el dispositivo en la sección de [Descargas](#). El paquete Software descargado se compone de la API de conexión y del controlador para el Sistema Operativo.

Para la correcta instalación, la carpeta gUSBampAPI debe estar en dentro de la carpeta de la plataforma Software, de esta forma se tiene todo en una localización determinada y se evitará las incompatibilidades con futuras versiones de la propia API. Los comandos son los siguientes:

1. `cd gUSBampAPI_1_16_01`
2. `chmod +777 install.sh`
3. `sudo ./install.sh`
4. `whereis gAPI.h`
5. COMPROBACIÓN ->gAPI: /usr/include/gAPI.h

Una vez instalada la API y el controlador vamos a configurar los ficheros para que serán llamados de la propia plataforma.

1. `cd gUSBamp_Demo_1_16_01`
2. `sudo apt-get install libboost-all-dev`
3. `make`
4. `./gUSBampAPI_Demo`
5. `cd ..`
6. `mv gUSBampAPI_Demo gUSBamp_files`

Llegados a este punto para los ficheros que deben estar en la carpeta gUSBamp\_files son los siguientes:

- data\_master.bin (archivo que se genera en la grabación.)
- gUSBampAPI\_Demo (ejecutable)
- main.cpp
- main.o
- Makefile

### B.1.2 Python 3.6.8

Para instalar esta versión de Python procederemos a realizar una serie de pasos válidos únicamente si en el ordenador a utilizar tiene el sistema operativo Ubuntu 16.04, 16.10 o 17.04. Pasos a seguir:

- `sudo apt install software-properties-common`
- `sudo add-apt-repository ppa:jonathonf/python-3.6`
- `sudo apt update`
- `sudo apt install python3.6`
- `sudo apt install python3.6-dev`
- `sudo apt install python3.6-venv`
- `wget https://bootstrap.pypa.io/get-pip.py`
- `sudo python3.6 get-pip.py`

Hay que tener especial atención si se dispone de otras versiones de Python instalada. Por lo general los cambios significativos en este lenguaje de programación hacen que cambie el nombre de cómo se ejecuta, por ejemplo, existen Python, Python3, Python3.6 y sucesivos.

### B.1.3 Librerías necesarias para la plataforma

La plataforma Software que se ha construido para la ejecución de este trabajo requiere de varias librerías. Algunas de las cuales requerirán de otras previas de forma indirecta, pero que se resumen en los siguientes pasos:

- `sudo apt install python3-pip`
- `pip3 --version`
- `sudo pip3 install pyserial`
- `sudo pip3 install numpy`
- `sudo pip3 install scipy==1.1.0`
- `sudo pip3 install PyQt5`
- `sudo apt-get install python3-matplotlib`
- `sudo pip3 install scikit-learn`
- `sudo pip3 install more-itertools`
- `sudo pip3 install --upgrade matplotlib`

Como se puede ver usamos el comando `pip3`, que es una versión de `pip` para Python 3. Este comando permite resolver y descargar todas aquellas librerías extra que necesita el paquete que estemos instalando, es muy útil y recomendable usarlo. **La versión de la Scipy debe ser la 1.1.0**, en caso contrario dará fallos muy poco intuitivos y provocará fallos en la aplicación.